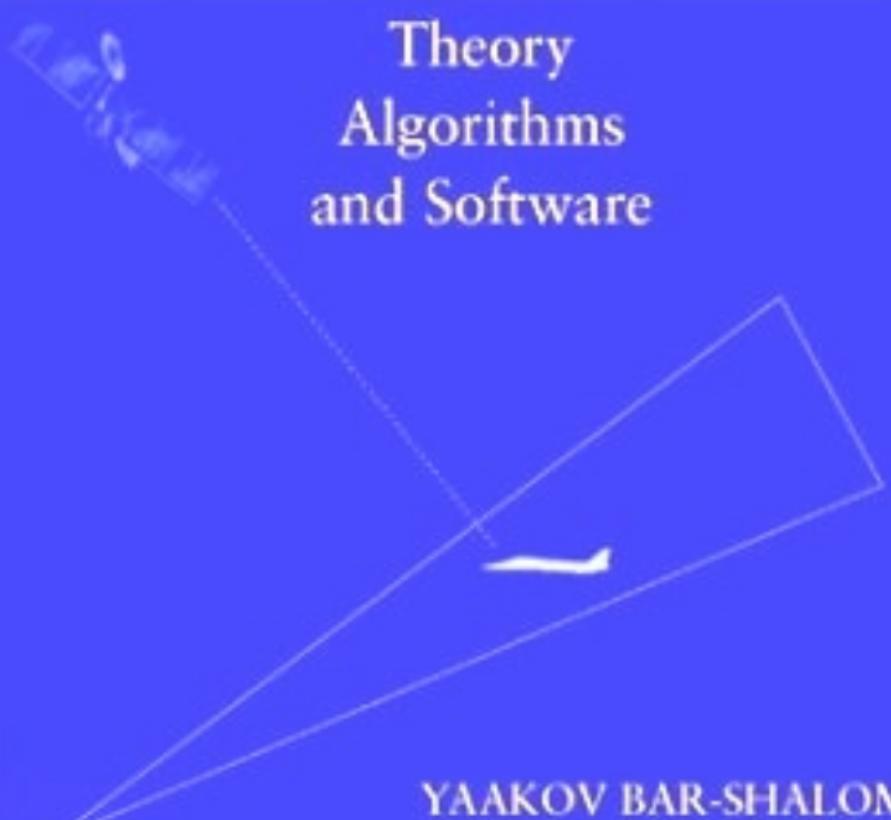


# Estimation with Applications to Tracking and Navigation

Theory  
Algorithms  
and Software



YAAKOV BAR-SHALOM  
X. RONG LI  
THIAGALINGAM KIRUBARAJAN

# **Estimation with Applications To Tracking and Navigation**

# **Estimation with Applications To Tracking and Navigation**

**Yaakov Bar-Shalom**

**X.-Rong Li**

**Thiagalingam Kirubarajan**



**A Wiley-Interscience Publication**

**JOHN WILEY & SONS, INC.**

**New York • Chichester • Weinheim • Brisbane • Singapore • Toronto**

This text is printed on acid-free paper. ©

Copyright © 2001 by John Wiley & Sons, Inc.

All rights reserved. Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4744. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 605 Third Avenue, New York, NY 10158-0012, (212) 850-6011, fax (212) 850-6008, E-Mail: PERMREQ @ WILEY.COM.

For ordering and customer service, call 1-800-CALL-WILEY.

*Library of Congress Cataloging-in-Publication Data*

Bar-Shalom, Yaakov.

Estimation with applications to tracking and navigation / by Yaakov Bar-Shalom,  
X.-Rong Li, Thiagalingam Kirubarajan.

p. cm.

Includes bibliographical references and index.

ISBN 0-471-41655-X (cloth)

1. Motion control devices. 2. Remote control. 3. Telecommunication systems. 4. Robots—Control systems. 5. Process control. 6. Estimation theory. I. Li, X.-Rong. II. Kirubarajan, Thiagalingam. III. Title.

TJ214.5 .B37 2001

681'.2—dc21

2001022366

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

To Eva, Tali, Yael and Michael

YBS

To Peizhu, Helen and Linda

XRL

To Appa, Amma, Ketha, Abi and Arun

TK

*Lemma 1.* Make things as simple as possible but not simpler.

A. Einstein

*Theorem 1.* By making things absolutely clear,  
people will become confused.

A Chinese fortune cookie

*Corollary 1.* We will make things  
simple  
but not too simple,  
clear  
but not too clear.

*Lemma 2.* Uncertainty is everywhere.

*Theorem 2.* Uncertainty cannot be conquered.

*Corollary 2.* Embrace it!

Paraphrased after Michael Moschen,  
professional juggler.

# Contents

PREFACE	xvii
ACRONYMS	xxi
MATHEMATICAL NOTATIONS	xxii
1 INTRODUCTION	1
1.1 BACKGROUND	1
1.1.1 Estimation and Related Areas	1
1.1.2 Applications of Estimation	3
1.1.3 Preview of Estimation/Filtering	4
1.1.4 An Example of State Estimation: Vehicle Collision Avoidance	10
1.2 SCOPE OF THE TEXT	15
1.2.1 Objectives	15
1.2.2 Overview and Chapter Prerequisites	16
1.3 BRIEF REVIEW OF LINEAR ALGEBRA AND LINEAR SYSTEMS	19
1.3.1 Definitions and Notations	19
1.3.2 Some Linear Algebra Operations	20
1.3.3 Inversion and the Determinant of a Matrix	21
1.3.4 Orthogonal Projection of Vectors	23
1.3.5 The Gradient, Jacobian and Hessian	24
1.3.6 Eigenvalues, Eigenvectors, and Quadratic Forms	25
1.3.7 Continuous-Time Linear Dynamic Systems — Controllability and Observability	27
1.3.8 Discrete-Time Lincar Dynamic Systems — Controllability and Observability	29
1.4 BRIEF REVIEW OF PROBABILITY THEORY	31
1.4.1 Events and the Axioms of Probability	31
1.4.2 Random Variables and Probability Density Function	33
1.4.3 Probability Mass Function	35
1.4.4 Mixed Random Variable and Mixed Probability-PDF	36
1.4.5 Expectations and Moments of a Scalar Random Variable	37
1.4.6 Joint PDF of Two Random Variables	38
1.4.7 Independent Events and Independent Random Variables	41
1.4.8 Vector-Valued Random Variables and Their Moments	41
1.4.9 Conditional Probability and PDF	44
1.4.10 The Total Probability Theorem	45
1.4.11 Bayes' Formula	47
1.4.12 Conditional Expectations and Their Smoothing Property	50
1.4.13 Gaussian Random Variables	51
1.4.14 Joint and Conditional Gaussian Random Variables	52
1.4.15 Expected Value of Quadratic and Quartic Forms	54
1.4.16 Mixture Probability Density Functions	55
1.4.17 Chi-Square Distributed Random Variables	57
1.4.18 Weighted Sum of Chi-Square Random Variables	60
1.4.19 Random Processes	61
1.4.20 Random Walk and the Wiener Process	65
1.4.21 Markov Processes	66
1.4.22 Random Sequences, Markov Sequences and Markov Chains	69

## X CONTENTS

1.4.23	The Law of Large Numbers and the Central Limit Theorem	70
1.5	BRIEF REVIEW OF STATISTICS	72
1.5.1	Hypothesis Testing	72
1.5.2	Confidence Regions and Significance	74
1.5.3	Monte Carlo Runs and Comparison of Algorithms	79
1.5.4	Tables of the Chi-Square and Gaussian Distributions	82
1.6	NOTES AND PROBLEMS	85
1.6.1	Bibliographical Notes	85
1.6.2	Problems	85
2	BASIC CONCEPTS IN ESTIMATION	89
2.1	INTRODUCTION	89
2.1.1	Outline	89
2.1.2	Basic Concepts – Summary of Objectives	89
2.2	THE PROBLEM OF PARAMETER ESTIMATION	90
2.2.1	Definitions	90
2.2.2	Models for Estimation of a Parameter	91
2.3	MAXIMUM LIKELIHOOD AND MAXIMUM A POSTERIORI ESTIMATORS	92
2.3.1	Definitions of ML and MAP Estimators	92
2.3.2	MLE vs. MAP Estimator with Gaussian Prior	92
2.3.3	MAP Estimator with One-Sided Exponential Prior	94
2.3.4	MAP Estimator with Diffuse Prior	95
2.3.5	The Sufficient Statistic and the Likelihood Equation	96
2.4	LEAST SQUARES AND MINIMUM MEAN SQUARE ERROR ESTIMATION	98
2.4.1	Definitions of LS and MMSE Estimators	98
2.4.2	Some LS Estimators	100
2.4.3	MMSE vs. MAP Estimator in Gaussian Noise	100
2.5	UNBIASED ESTIMATORS	101
2.5.1	Definition	101
2.5.2	Unbiasedness of an ML and a MAP Estimator	102
2.5.3	Bias in the ML Estimation of Two Parameters	102
2.6	THE VARIANCE AND MSE OF AN ESTIMATOR	104
2.6.1	Definitions of Estimator Variances	104
2.6.2	Comparison of Variances of an ML and a MAP Estimator	105
2.6.3	The Variances of the Sample Mean and Sample Variance	106
2.6.4	Estimation of the Probability of an Event	107
2.7	CONSISTENCY AND EFFICIENCY OF ESTIMATORS	108
2.7.1	Consistency	108
2.7.2	The Cramer-Rao Lower Bound and the Fisher Information Matrix	109
2.7.3	Proof of the Cramer-Rao Lower Bound	110
2.7.4	An Example of Efficient Estimator	112
2.7.5	Large Sample Properties of the ML Estimator	113
2.8	SUMMARY	114
2.8.1	Summary of Estimators	114
2.8.2	Summary of Estimator Properties	115
2.9	NOTES AND PROBLEMS	115
2.9.1	Bibliographical Notes	115
2.9.2	Problems	116
3	LINEAR ESTIMATION IN STATIC SYSTEMS	121
3.1	INTRODUCTION	121
3.1.1	Outline	121
3.1.2	Linear Estimation in Static Systems — Summary of Objectives	121
3.2	ESTIMATION OF GAUSSIAN RANDOM VECTORS	122
3.2.1	The Conditional Mean and Covariance for Gaussian Random Vectors	122
3.2.2	Estimation of Gaussian Random Vectors — Summary	123
3.3	LINEAR MINIMUM MEAN SQUARE ERROR ESTIMATION	123
3.3.1	The Principle of Orthogonality	123

3.3.2	Linear MMSE Estimation for Vector Random Variables	127
3.3.3	Linear MMSE Estimation — Summary	129
3.4	LEAST SQUARES ESTIMATION	129
3.4.1	The Batch LS Estimation	129
3.4.2	The Recursive LS Estimator	132
3.4.3	Examples and Incorporation of Prior Information	135
3.4.4	Nonlinear LS — An Example	137
3.4.5	LS Estimation — Summary	145
3.5	POLYNOMIAL FITTING	146
3.5.1	Fitting a First-Order Polynomial to Noisy Measurements	146
3.5.2	Fitting a General Polynomial to a Set of Noisy Measurements	149
3.5.3	Mapping of the Estimates to an Arbitrary Time	152
3.5.4	Polynomial Fitting — Summary	154
3.6	GOODNESS-OF-FIT AND STATISTICAL SIGNIFICANCE OF PARAMETER ESTIMATES	154
3.6.1	Hypothesis Testing Formulation of the Problem	154
3.6.2	The Fitting Error in a Least Squares Estimation Problem	156
3.6.3	A Polynomial Fitting Example	159
3.6.4	Order Selection in Polynomial Fitting — Summary	161
3.7	USE OF LS FOR A NONLINEAR PROBLEM: BEARINGS-ONLY TARGET MOTION ANALYSIS	161
3.7.1	The Problem	161
3.7.2	Observability of the Target Parameter in Passive Localization	162
3.7.3	The Likelihood Function for Target Parameter Estimation	163
3.7.4	The Fisher Information Matrix for the Target Parameter	164
3.7.5	The Goodness-of-Fit Test	167
3.7.6	Testing for Efficiency with Monte Carlo Runs	168
3.7.7	A Localization Example	169
3.7.8	Passive Localization — Summary	169
3.8	NOTES, PROBLEMS AND A PROJECT	172
3.8.1	Bibliographical Notes	172
3.8.2	Problems	172
3.8.3	PROJECT: An Interactive Program for Bearings-Only Target Localization	176
4	LINEAR DYNAMIC SYSTEMS WITH RANDOM INPUTS	179
4.1	INTRODUCTION	179
4.1.1	Outline	179
4.1.2	Linear Stochastic Systems — Summary of Objectives	179
4.2	CONTINUOUS-TIME LINEAR STOCHASTIC DYNAMIC SYSTEMS	180
4.2.1	The Continuous-Time State-Space Model	180
4.2.2	Solution of the Continuous-Time State Equation	181
4.2.3	The State as a Markov Process	183
4.2.4	Propagation of the State's Mean and Covariance	184
4.2.5	Frequency Domain Approach	185
4.3	DISCRETE-TIME LINEAR STOCHASTIC DYNAMIC SYSTEMS	187
4.3.1	The Discrete-Time State-Space Model	187
4.3.2	Solution of the Discrete-Time State Equation	189
4.3.3	The State as a Markov Process	190
4.3.4	Propagation of the State's Mean and Covariance	191
4.3.5	Frequency Domain Approach	192
4.4	SUMMARY	195
4.4.1	Summary of State Space Representation	195
4.4.2	Summary of Prewhitenning	195
4.5	NOTES AND PROBLEMS	196
4.5.1	Bibliographical Notes	196
4.5.2	Problems	196

5	STATE ESTIMATION IN DISCRETE-TIME LINEAR DYNAMIC SYSTEMS	199
5.1	INTRODUCTION	199
5.1.1	Outline	199
5.1.2	Discrete-Time Linear Estimation — Summary of Objectives	199
5.2	LINEAR ESTIMATION IN DYNAMIC SYSTEMS — THE KALMAN FILTER	200
5.2.1	The Dynamic Estimation Problem	200
5.2.2	Dynamic Estimation as a Recursive Static Estimation	202
5.2.3	Derivation of the Dynamic Estimation Algorithm	204
5.2.4	Overview of the Kalman Filter Algorithm	207
5.2.5	The Matrix Riccati Equation	211
5.2.6	Properties of the Innovations and the Likelihood Function of the System Model	213
5.2.7	The Innovations Representation	214
5.2.8	Some Orthogonality Properties	215
5.2.9	The Kalman Filter — Summary	215
5.3	EXAMPLE OF A FILTER	218
5.3.1	The Model	218
5.3.2	Results for a Kalman Filter	219
5.3.3	A Step-by-Step Demonstration of DynaEst™	219
5.4	CONSISTENCY OF STATE ESTIMATORS	232
5.4.1	The Problem of Filter Consistency	232
5.4.2	Definition and the Statistical Tests for Filter Consistency	234
5.4.3	Examples of Filter Consistency Testing	237
5.4.4	Absolute Errors	243
5.4.5	Filter Consistency — Summary	244
5.5	INITIALIZATION OF STATE ESTIMATORS	245
5.5.1	Initialization and Consistency	245
5.5.2	Initialization in Simulations	246
5.5.3	A Practical Implementation in Tracking	247
5.5.4	Filter Initialization — Summary	248
5.6	SENSITIVITY	248
5.6.1	Model Mismatch	249
5.6.2	Reduced-Order Filters	254
5.6.3	Suboptimal Gains	256
5.6.4	Examples of Modeling Errors and Filter Approximations	256
5.7	NOTES AND PROBLEMS	261
5.7.1	Bibliographical Notes	261
5.7.2	Problems	261
5.7.3	Computer Applications	265
6	ESTIMATION FOR KINEMATIC MODELS	267
6.1	INTRODUCTION	267
6.1.1	Outline	267
6.1.2	Kinematic Models — Summary of Objectives	267
6.2	DISCRETIZED CONTINUOUS-TIME KINEMATIC MODELS	268
6.2.1	The Kinematic Models	268
6.2.2	Continuous White Noise Acceleration Model	269
6.2.3	Continuous Wiener Process Acceleration Model	270
6.3	DIRECT DISCRETE-TIME KINEMATIC MODELS	272
6.3.1	Introduction	272
6.3.2	Discrete White Noise Acceleration Model	273
6.3.3	Discrete Wiener Process Acceleration Model	274
6.3.4	Kinematic Models — Summary	275
6.4	EXPLICIT FILTERS FOR NOISELESS KINEMATIC MODELS	276
6.4.1	LS Estimation for Noiseless Kinematic Models	276
6.4.2	The KF for Noiseless Kinematic Models	276
6.5	STEADY-STATE FILTERS FOR NOISY KINEMATIC MODELS	277
6.5.1	The Problem	277
6.5.2	Derivation Methodology for the Alpha-Beta Filter	278

6.5.3	The Alpha-Beta Filter for the DWNA Model	280
6.5.4	The Alpha-Beta Filter for the Discretized CWNA Model	286
6.5.5	The Alpha-Beta-Gamma Filter for the DWPA Model	289
6.5.6	A System Design Example for Sampling Rate Selection	292
6.5.7	Alpha-Beta and Alpha-Beta-Gamma Filters — Summary	293
6.6	NOTES AND PROBLEMS	294
6.6.1	Bibliographical Notes	294
6.6.2	Problems	295
7	COMPUTATIONAL ASPECTS OF ESTIMATION	301
7.1	INTRODUCTION	301
7.1.1	Implementation of Linear Estimation	301
7.1.2	Outline	302
7.1.3	Computational Aspects — Summary of Objectives	303
7.2	THE INFORMATION FILTER	303
7.2.1	Recursions for the Information Matrices	303
7.2.2	Overview of the Information Filter Algorithm	306
7.2.3	Recursion for the Information Filter State	307
7.3	SEQUENTIAL PROCESSING OF MEASUREMENTS	308
7.3.1	Block vs. Sequential Processing	308
7.3.2	The Sequential Processing Algorithm	309
7.4	SQUARE-ROOT FILTERING	311
7.4.1	The Steps in Square-Root Filtering	311
7.4.2	The $LDL'$ Factorization	312
7.4.3	The Predicted State Covariance	312
7.4.4	The Filter Gain and the Updated State Covariance	314
7.4.5	Overview of the Square Root Sequential Scalar Update Algorithm	315
7.4.6	The Gram-Schmidt Orthogonalization Procedure	315
7.5	NOTES AND PROBLEMS	317
7.5.1	Bibliographical Notes	317
7.5.2	Problems	317
8	EXTENSIONS OF DISCRETE-TIME LINEAR ESTIMATION	319
8.1	INTRODUCTION	319
8.1.1	Outline	319
8.1.2	Extensions of Estimation — Summary of Objectives	319
8.2	AUTOCORRELATED PROCESS NOISE	320
8.2.1	The Autocorrelated Process Noise Problem	320
8.2.2	An Exponentially Autocorrelated Noise	321
8.2.3	The Augmented State Equations	322
8.2.4	Estimation with Autocorrelated Process Noise — Summary	324
8.3	CROSS-CORRELATED MEASUREMENT AND PROCESS NOISE	324
8.3.1	Cross-Correlation at the Same Time	324
8.3.2	Cross-Correlation One Time Step Apart	326
8.3.3	State Estimation with Decorrelated Noise Sequences — Summary	327
8.4	AUTOCORRELATED MEASUREMENT NOISE	327
8.4.1	Whitening of the Measurement Noise	327
8.4.2	The Estimation Algorithm with the Whitened Measurement Noise	329
8.4.3	Autocorrelated Measurement Noise — Summary	330
8.5	PREDICTION	330
8.5.1	Types of Prediction	330
8.5.2	The Algorithms for the Different Types of Prediction	330
8.5.3	Prediction — Summary	332
8.6	SMOOTHING	333
8.6.1	Types of Smoothing	333
8.6.2	Fixed-Interval Smoothing	334
8.6.3	Overview of Smoothing	337
8.6.4	Smoothing — Summary	338

**xiv CONTENTS**

8.7	NOTES AND PROBLEMS	338
8.7.1	Bibliographical Notes	338
8.7.2	Problems	338
9	CONTINUOUS-TIME LINEAR STATE ESTIMATION	341
9.1	INTRODUCTION	341
9.1.1	Outline	341
9.1.2	Continuous-Time Estimation — Summary of Objectives	341
9.2	THE CONTINUOUS-TIME LINEAR STATE ESTIMATION FILTER	342
9.2.1	The Continuous-Time Estimation Problem	342
9.2.2	Connection Between Continuous - and Discrete-Time Representations and Their Noise Statistics	343
9.2.3	The Continuous-Time Filter Equations	345
9.2.4	The Continuous-Time Innovation	347
9.2.5	Asymptotic Properties of the Continuous-Time Riccati Equation	349
9.2.6	Examples of Continuous-Time Filters	351
9.2.7	Overview of the Kalman-Bucy Filter	353
9.2.8	Continuous-Time State Estimation — Summary	354
9.3	PREDICTION AND THE CONTINUOUS-DISCRETE FILTER	355
9.3.1	Prediction of the Mean and Covariance	355
9.3.2	The Various Types of Prediction	356
9.3.3	The Continuous-Discrete Filter	357
9.4	DUALITY OF ESTIMATION AND CONTROL	358
9.4.1	The Two Problems	358
9.4.2	The Solutions to the Estimation and the Control Problems	359
9.4.3	Properties of the Solutions	360
9.5	THE WIENER-HOPF PROBLEM	362
9.5.1	Formulation of the Problem	362
9.5.2	The Wiener-Hopf Equation	362
9.6	NOTES AND PROBLEMS	366
9.6.1	Bibliographical Notes	366
9.6.2	Problems	367
10	STATE ESTIMATION FOR NONLINEAR DYNAMIC SYSTEMS	371
10.1	INTRODUCTION	371
10.1.1	Outline	371
10.1.2	Nonlinear Estimation — Summary of Objectives	371
10.2	ESTIMATION IN NONLINEAR STOCHASTIC SYSTEMS	372
10.2.1	The Model	372
10.2.2	The Optimal Estimator	373
10.2.3	Proof of the Recursion of the Conditional Density of the State	374
10.2.4	Example of Linear vs. Nonlinear Estimation of a Parameter	376
10.2.5	Estimation in Nonlinear Systems with Additive Noise	379
10.2.6	Optimal Nonlinear Estimation — Summary	381
10.3	THE EXTENDED KALMAN FILTER	381
10.3.1	Approximation of the Nonlinear Estimation Problem	381
10.3.2	Derivation of the EKF	383
10.3.3	Overview of the EKF Algorithm	385
10.3.4	An Example: Tracking with an Angle-Only Sensor	387
10.3.5	The EKF — Summary	394
10.4	ERROR COMPENSATION IN LINEARIZED FILTERS	395
10.4.1	Some Heuristic Methods	395
10.4.2	An Example of Use of the Fudge Factor	396
10.4.3	An Example of Debiasing: Conversion from Polar to Cartesian	397
10.4.4	Error Compensation in Linearized Filters — Summary	402
10.5	SOME ERROR REDUCTION METHODS	404
10.5.1	Improved State Prediction	404
10.5.2	The Iterated Extended Kalman Filter	404

10.6	MAXIMUM A POSTERIORI TRAJECTORY ESTIMATION VIA DYNAMIC PROGRAMMING	407
10.6.1	The Approach	407
10.6.2	The Dynamic Programming for Trajectory Estimation	408
10.7	NONLINEAR CONTINUOUS-DISCRETE FILTER	409
10.7.1	The Model	409
10.7.2	The Fokker-Planck Equation	410
10.7.3	Example	413
10.8	NOTES, PROBLEMS AND A PROJECT	414
10.8.1	Bibliographical Notes	414
10.8.2	Problems	414
10.8.3	Project — Nonlinear Filtering with Angle-Only Measurements	419
11	ADAPTIVE ESTIMATION AND MANEUVERING TARGETS	421
11.1	INTRODUCTION	421
11.1.1	Adaptive Estimation — Outline	421
11.1.2	Adaptive Estimation — Summary of Objectives	423
11.2	ADJUSTABLE LEVEL PROCESS NOISE	424
11.2.1	Continuous Noise Level Adjustment	424
11.2.2	Process Noise with Several Discrete Levels	424
11.2.3	Adjustable Level Process Noise — Summary	426
11.3	INPUT ESTIMATION	427
11.3.1	The Model	427
11.3.2	The Innovations as a Linear Measurement of the Unknown Input	428
11.3.3	Estimation of the Unknown Input	429
11.3.4	Correction of the State Estimate	430
11.3.5	Input Estimation — Summary	431
11.4	THE VARIABLE STATE DIMENSION APPROACH	431
11.4.1	The Approach	431
11.4.2	The Maneuver Detection and Model Switching	432
11.4.3	Initialization of the Augmented Model	433
11.4.4	VSD Approach — Summary	434
11.5	A COMPARISON OF ADAPTIVE ESTIMATION METHODS FOR MANEUVERING TARGETS	435
11.5.1	The Problem	435
11.5.2	The White Noise Model with Two Levels	436
11.5.3	The IE and VSD Methods	436
11.5.4	Statistical Test for Comparison of the IE and VSD Methods	438
11.5.5	Comparison of Several Algorithms — Summary	440
11.6	THE MULTIPLE MODEL APPROACH	441
11.6.1	Formulation of the Approach	441
11.6.2	The Static Multiple Model Estimator	441
11.6.3	The Dynamic Multiple Model Estimator	444
11.6.4	The GPB1 Multiple Model Estimator for Switching Models	447
11.6.5	The GPB2 Multiple Model Estimator for Switching Models	449
11.6.6	The Interacting Multiple Model Estimator	453
11.6.7	An Example with the IMM Estimator	457
11.6.8	Use of DynaEst <sup>TM</sup> to Design an IMM Estimator	460
11.6.9	The Multiple Model Approach — Summary	465
11.7	DESIGN OF AN IMM ESTIMATOR FOR ATC TRACKING	466
11.7.1	ATC Motion Models	466
11.7.2	The EKF for the Coordinated Turn Model	468
11.7.3	Selection of Models and Parameters	470
11.7.4	The ATC Scenario	471
11.7.5	Results and Discussion	472
11.8	WHEN IS AN IMM ESTIMATOR NEEDED?	476
11.8.1	Kalman Filter vs. IMM Estimator	477

## **xvi CONTENTS**

11.9	USE OF EKF FOR SIMULTANEOUS STATE AND PARAMETER ESTIMATION	481
11.9.1	Augmentation of the State	481
11.9.2	An Example of Use of the EKF for Parameter Estimation	482
11.9.3	EKF for Parameter Estimation — Summary	484
11.10	NOTES, PROBLEMS, AND TERM PROJECT	484
11.10.1	Bibliographical Notes	484
11.10.2	Problems	485
11.10.3	Term Project — IMM Estimator for Air Traffic Control	488
12	INTRODUCTION TO NAVIGATION APPLICATIONS	491
12.1	INTRODUCTION	491
12.1.1	Navigation Applications — Outline	491
12.1.2	Navigation Applications — Summary of Objectives	492
12.2	COMPLEMENTARY FILTERING FOR NAVIGATION	492
12.2.1	The Operation of Complementary Filtering	492
12.2.2	The Implementation of Complementary Filtering	493
12.3	INERTIAL NAVIGATION SYSTEMS	495
12.4	MODELS FOR INERTIAL NAVIGATION SYSTEMS	496
12.4.1	State Models	496
12.4.2	Sensor Error Models	496
12.4.3	Single-Axis Models	497
12.4.4	Three-Axis Models	499
12.4.5	Coordinate Transformation	500
12.5	THE GLOBAL POSITIONING SYSTEM (GPS)	501
12.5.1	The GPS Segments	502
12.5.2	GPS Satellite Constellation	502
12.6	GPS POSITIONING	502
12.6.1	The GPS Principle	502
12.6.2	The GPS Signals	503
12.6.3	The GPS Observables	506
12.7	THE ACCURACY OF GPS POSITIONING	507
12.7.1	Dilution of Precision	507
12.7.2	GPS Positioning Accuracy	509
12.8	STATE-SPACE MODELS FOR GPS	511
12.8.1	Models for Receiver Clock State	511
12.8.2	Dynamic Models	512
12.8.3	Linearized Measurement Model	512
12.8.4	A Model for Exponentially Autocorrelated Noise	513
12.8.5	Coordinate Transformation	515
12.9	EXAMPLE: GPS NAVIGATION WITH IMM ESTIMATOR	515
12.9.1	Generation of Satellite Trajectories	516
12.9.2	Generation of Trajectories and Pseudorange Measurements	517
12.9.3	State-Space Models	518
12.9.4	Simulation Results and Discussion	520
12.9.5	Do We Need an IMM Estimator for GPS?	523
12.10	INTEGRATED NAVIGATION	523
12.10.1	Integration by Complementary Filtering	524
12.10.2	Example	525
12.10.3	Integration by Centralized Estimation Fusion	527
12.10.4	Integration by Distributed Estimation Fusion	528
12.11	NOTES AND PROBLEMS	530
12.11.1	Bibliographical Notes	530
12.11.2	Problems	530
12.11.3	Term Project — Extended Kalman Filter for GPS	533
	BIBLIOGRAPHY	537
	INDEX	547

# PREFACE

This text, which also doubles as a *set of lecture notes available in viewgraph version* (downloadable as detailed below), presents the material from a graduate-level course on *Theory and Computational Algorithms for Estimation* offered in the Department of Electrical and Computer Engineering at the University of Connecticut. This course is a standard requirement in the department's M.S. program in Information, Communication and Decision Systems and is meant for second-semester graduate students. The prerequisites are a solid knowledge of linear systems and probability theory at the first-semester graduate level. These, as well as some additional useful material from Statistics, are summarized for ready reference in Chapter 1 to enable readers to acquire the necessary background or review it. This makes the text *completely self-contained* and accessible for a person with a typical B.S.E.E. degree or equivalent. This text is a major revision and expansion of Y. Bar-Shalom and X. R. Li, **Estimation and Tracking: Principles, Techniques, and Software**, originally published by Artech House, 1993 (©YBS Publishing, 1997).

The main goal is to convey the knowledge necessary for the *design and evaluation of state estimation algorithms that operate in a stochastic environment*. These algorithms form the backbone of **information extraction systems** for **remote sensing of moving objects** as well as other similar applications. Since **information fusion** — combination of information from several sources — is an area with significant recent activity, it should be kept in mind that information extraction is a crucial step before fusion. Furthermore, since fusion should be done accounting for the existing uncertainties, it is important to quantify the accuracies of the information to be fused.

This book also provides opportunity for sharpening the students' analytical skills in addition to using software. While the greatest recent advances have been in computation, the quality of a software is (in a simplistic manner) the product of the software skills and algorithmic skills that went into its development. The algorithm development requires analytical skills, which, unfortunately, have become too scarce lately.

While the Kalman filter (KF) is the best-known state estimator, it is neither the only one nor the most versatile. The material, which covers a number of practical estimation algorithms (as many as it is felt practical for a semester-long course), leads to a major design project that is unique to this text — it combines the basics of standard estimation with the latest adaptive multiple model techniques, which use KFs as modules. The emphasis is on mapping the physical properties of the object of interest (typically called "target," whether friendly or not) and of the sensors — e.g., an aircraft in a radar-based air traffic control tracking system — into the parameters of the mathematical model, namely, the statistical characterization of the random processes describing the uncertainties of this problem. As a prelude to state estimation, parameter estimation is discussed in sufficient detail to convey the concept of "limit of extractable information" in a system.

The approach is a balanced combination of mathematics — linear systems and probability theory — in order to understand how a state estimator should be designed, with the necessary tools from statistics in order to interpret the results. The use of statistical techniques has been somewhat neglected in the engineering literature pertaining to state estimation, but it is necessary for (the nontrivial task of) interpreting stochastic data and answering the question whether a design can be accepted as “good.” This is particularly important for practicing engineers and is presented in sufficient detail based on our belief (and extensive experience with real systems) that it should be an integral part of advanced engineering education.<sup>1</sup>

The material covers the topics usually taught in control-oriented EE/systems and aeronautical engineering programs. The relevance extends to other areas dealing with control in mechanical or chemical engineering. Recently, the state estimation techniques have been gaining wider attention due to their applicability to such fields as robotics, computer vision for autonomous navigation, and image feature extraction with application to medical diagnosis. While the course is mainly directed toward the M.S. students, it is also part of the Ph.D. program at the University of Connecticut, with the intent of providing the students with the knowledge to tackle real-world problems, whether by using existing algorithms or by developing new ones.<sup>2</sup>

The presentation of the material stresses the algorithms, their properties and the understanding of the assumptions behind them. We do not subscribe to the philosophy of “Give me the facts and don’t bother me with details.” Consequently, proofs are given to the extent that they are relevant to understanding the results. This is intended to be a modest step in bridging the much talked about “gap between theory and practice” — it will illustrate to students the usefulness of state estimation for the real world and provide to engineers and scientists working in industry or laboratories a broader understanding of the algorithms used in practice. It might also avoid the situation summarized by a participant at one of the continuing education courses taught by the first author as follows: “Although I studied Kalman filters when I worked toward my Ph.D. (at one of the major U.S. universities), I did not expect that they worked with real data.” This happens when, because of the theorems, the students cannot see the forest of applications.

Tuning of a KF — the choice of its design parameters — is an art. One of the contributions of this text is to make it less of a black magic technique and more of a systematic approach by connecting the filter parameters to physical system parameters, namely, the object motion uncertainty — its predictability — and the sensor measurement uncertainty — the sensor errors. This is particularly important when KFs are used as modules in an adaptive estimator, like the Interacting Multiple Model (IMM) estimator. Another contribution is providing guidelines as to what estimator should be used in specific problems, namely, when an (adaptive) IMM estimator is preferable over a KF.

## The Text and Hyperlinked Viewgraph Format

The format of this text is also unique — *textgraph*<sup>TM</sup> — in that it dares to attempt to accomplish two goals in one format: to serve as a self-contained *concise text*, without excess verbosity, and, at the same time, to enable the lecturer to use the pages of this text as *viewgraphs for lectures*. To this purpose, a double-spaced version of all the pages of this text (with appropriate page breaks)

---

<sup>1</sup>The authors disclaim any responsibility for severe damage readers might suffer when falling asleep face forward while reading this book.

<sup>2</sup>As Ben Franklin said, the goal in his life was “to rise to an (sic) happy mediocrity and secure competency.” Our objective is to provide the tools for future successful applications.

is available for instructors in the form of a pdf file at <http://esplab1.ee.uconn.edu/estvu.zip> with the password for downloading obtainable (upon the adoption of this book as a course text) from [ybs@ee.uconn.edu](mailto:ybs@ee.uconn.edu). Instructors can prepare transparencies for conventional projector viewgraphs, or the pdf file can be used for *direct electronic projection*. For projection from a computer, an additional feature is available: hyperlinks in the pdf file for all cross-references (equations, figures, tables, sections, bibliographical items). Starred sections (or subsections) can be skipped in lecturing at the discretion of the lecturer — by clicking on the star the hypertext display will automatically go to the end of the corresponding section, marked by a box.

Experience over a number of years shows that students with a copy of these notes in front of them can concentrate on the ideas, rather than having to copy a copious amount of equations from the board. Some of the graduate students who took the Estimation course taught using this *textgraph*<sup>TM</sup> even complained that other graduate courses had not adopted this instruction method.

This format is also a way of making lectures more efficient since it makes it possible to convey more material than through the standard lecturing technique where one has to write everything on the board. Last, but not least, the need to spend time on preparing lecture notes, as well as the main hazard of the teaching profession (the inhalation of chalk or marker dust), can be avoided. The ready availability of such class notes also makes it much easier to hold in-house courses in an industrial or government laboratory environment where senior staff — with knowledge of the material but limited time — can serve as lecturers.

It is our hope that this text will contribute to the education of *future practitioners*, as well as be of use to those already involved in the application of state estimation to real-world problems. This *textgraph*<sup>TM</sup> was written with the philosophy that engineers should be able to do more than run-of-the-mill work — they should be prepared to work at the frontiers of knowledge.<sup>3</sup>

## A Note on the Printing Style

For the convenience of the reader, all major concepts/terms, when they are introduced or used in a particular application, are in *italic boldface* and indexed. Also, all the index terms appear in the text in slanted boldface.

Italics are used for emphasized ideas/properties. The main equations of important algorithms are highlighted with boxes, and the algorithms are summarized in flowcharts.

## The Companion Software DynaEst<sup>TM</sup>

*DynaEst*<sup>TM</sup> is a completely self-explanatory menu-driven interactive design and evaluation tool for Kalman filters and an adaptive Interacting Multiple Model (IMM) estimator. The package provides the following important flexibilities as a design tool:

**Estimator design.** The user may specify any linear Gaussian true plant (dynamic) and measurement models, as well as the plant and measurement models for the estimator. The true and the filter-assumed models can be different. Models are valid for all dimensions. Reduced-order filters can be implemented.

**Selection of model parameters.** The user may specify any parameter of the true model and/or the filter-assumed model(s), such as the noise means and covariances.

<sup>3</sup>As the ancient Romans would have said: “Orchides forum trahite, cordes et mentes veniant.”

**Statistical testing.** Most relevant statistical tests of the Kalman filters may be performed, including the normalized estimation error squared (NEES), the normalized innovation squared (NIS), and the RMS errors.

**Capability of running on external data file.** Both internally generated data (simulation) or imported data can be used.

*DynaEst*<sup>TM</sup> is user friendly and has the following features:

- MATLAB based (v. 4.0 or higher, including the student edition)
- Self-explanatory menu
- Convenient user specification of parameters (on screen or in an online editable file)
- Printing on screen, to a printer, or to a file
- Automatic (and flexible) installation
- Access to the OS and MATLAB from within *DynaEst*<sup>TM</sup>
- Easily readable source code transparent to the user (and thus user-modifiable)
- Knowledge of MATLAB not required (but helpful, of course)
- For convenience, the *latest version* is available for the purchaser of this text at <http://esplab1.ce.uconn.edu/dynaest.zip>.

*DynaEst*<sup>TM</sup> provides valuable hands-on experience on Kalman filters and the state-of-the-art Interacting Multiple Model (IMM) estimator for any person who has access to MATLAB. With the help of *DynaEst*<sup>TM</sup>, the principles/concepts and techniques treated in this book can be better understood and remembered. Some practical problems of estimation and tracking can be handled by *DynaEst*<sup>TM</sup>. The source code of *DynaEst*<sup>TM</sup>, which is easy to read, provides a sample implementation of the Kalman filter, as well as the state-of-the-art IMM estimator.

*DynaEst*<sup>TM</sup> has been used to generate several of the examples in the text. Also, there are a number of Computer Application Exercises at the end of the chapters dealing with state estimation, designed to use *DynaEst*<sup>TM</sup>.

### System Requirements

- IBM PC compatible
- MATLAB (professional, education, or student version)
- An operating system capable of running MATLAB 4.0 or higher (e.g., WINDOWS 95/98/NT or SUN SOLARIS)

### Acknowledgements

Many thanks to all the colleagues, friends and former students who helped in putting together the material that eventually became this book: H. Blom, L. Campo, K. C. Chang, E. Daeipour, D. L. Kleinman, A. Kumar, C. Jauffret, D. Lerro, X. D. Lin, P. B. Luh, K. R. Pattipati, C. Rago, H. M. Shertukde, P. K. Willett and S. W. Yeom.

Also, thanks to the sponsors who have supported the research that led to this book: J. G. Smith and Dr. R. Madan from ONR, Dr. N. Glassman from AFOSR, Lt. Col. Dr. J. Myers from BMDO and Dr. R. K. Baheti from NSF.

# NOMENCLATURE

## ACRONYMS

ATC	air traffic control
cdf	cumulative (probability) distribution function
cpmf	cumulative probability mass function
CLT	central limit theorem
CRLB	Cramer-Rao lower bound
EKF	extended Kalman filter
FIM	Fisher information matrix
GPB	generalized pseudo-Bayesian
GPS	Global Positioning System
HOT	higher order terms
IE	input estimation
i.i.d.	independent identically distributed
IMM	interacting multiple model (estimator)
INS	inertial navigation system
KBF	Kalman-Bucy filter (continuous time)
KF	Kalman filter (discrete time)
LF	likelihood function
LG	linear-Gaussian (assumption)
LLN	law of large numbers
LMMSE	linear MMSE
LOS	line of sight
LS	least squares
LSE	least squares estimator (or estimate)
MAP	maximum a posteriori
ML	maximum likelihood
MLE	maximum likelihood estimator (or estimate)
MM	multiple model
MMSE	minimum mean square error
MSE	mean square error
NEES	normalized estimation error squared
NIS	normalized innovation squared
pdf	probability density function
pmf	probability mass function
RMS	root mean square
VSD	variable state dimension

## MATHEMATICAL NOTATIONS

$\arg \max$	argument that maximizes
$\arg \min$	argument that minimizes
$\chi_n^2$	chi-square distribution with $n$ degrees of freedom [Subsection 1.4.17]
$\chi_n^2(Q)$	100 $Q$ percentile point of chi-square distribution with $n$ degrees of freedom [Subsection 1.5.4]
$\text{col}(\cdot)$	column vector [Subsection 1.3.1]
$\text{cov}$	covariance
$\delta(\cdot)$	Dirac (continuous) delta function (impulse function)
$\delta_{ij}$	Kronecker (discrete) delta function
$\text{diag}(\cdot)$	diagonal or block-diagonal matrix [Subsection 1.3.3, Subsection 3.4.1]
$E[\cdot]$	expectation
$\mathcal{G}(Q)$	100 $Q$ percentile point of standard normal (Gaussian) distribution [Subsection 1.5.4]
$\Lambda$	likelihood function
$\mu(\cdot)$	pmf
$\mathcal{N}(x; \bar{x}, \sigma^2)$	pdf of a normal (Gaussian) random variable $x$ with mean $\bar{x}$ and variance $\sigma^2$ [Subsection 1.4.13]
$\sim (\bar{x}, \sigma^2)$	a random variable with mean $\bar{x}$ and variance $\sigma^2$ [Subsection 1.4.13]
$\mathcal{N}(\bar{x}, \sigma^2)$	normal (Gaussian) distribution with mean $\bar{x}$ and variance $\sigma^2$ [Subsection 1.4.13]
$\mathcal{N}(x; \bar{x}, P)$	pdf of a normal (Gaussian) random vector $x$ with mean $\bar{x}$ and covariance $P$ [Subsection 1.4.13]
$n_x$	dimension of the vector $x$
$O(\cdot)$	order of magnitude of
$p[\cdot]$	pdf [Subsection 1.4.2]
$p(\cdot)$	pdf [Subsection 1.4.2]
$P(k j)$	conditional covariance matrix of state at time $k$ given observations through time $j$
$P_{xz}$	covariance (matrix) of $x$ and $z$
$P_{x z}$	conditional covariance (matrix) of $x$ given $z$
$P[\cdot, \cdot]$	mixed joint probability-pdf
$P\{\cdot\}$	probability of an event
$\tilde{P}$	the spread of the means term [Subsection 1.4.16]
$P_D$	probability of detection [Subsection 1.5.1]
$P_{FA}$	false alarm probability [Subsection 1.5.1]
$\sigma_\theta$	standard deviation of $\theta$
$\sigma_\theta^2$	variance of $\theta$
$\text{tr}(\cdot)$	trace (of a matrix) [Subsection 1.3.2]
$\mathcal{U}(a, b)$	uniform distribution over the interval $[a, b]$
$\mathcal{U}(x; a, b)$	pdf of random variable $x$ uniformly distributed over the interval $[a, b]$ [Subsection 1.4.5]
$\text{var}$	variance
$Z^k$	the sequence $z(j)$ , $j = 1, \dots, k$ [Subsection 2.2.1]
$0$	scalar zero, zero (null) vector, zero (null) matrix
$\emptyset$	empty set, impossible event [Subsection 1.4.1]

$1$	unity, or a (column) vector with all elements unity
$1(\cdot)$	unit step function
$'$	transposition (of a matrix or a vector) [Subsection 1.3.1]
$(\cdot)^*$	complex conjugate and transposed [Subsection 1.4.23]
$\bar{x}$	expected value of $x$ [Subsection 1.4.5],
$\hat{x}$	estimate of $x$ [Subsection 2.2.1],
$\tilde{x}$	error corresponding to the estimate of $x$ [Subsection 2.2.1],
$\bar{A}$	complement of set (or event) $A$ [Subsection 1.4.1]
$\forall$	for all
$(\cdot  \cdot)$	conditioning (for probabilities, estimates or covariances)
$  \cdot  $	determinant (of a matrix)
$  \cdot  $	magnitude (of a scalar)
$\  \cdot \ $	norm of a vector [Subsection 1.3.4]
$\ \cdot\ $	norm of a matrix [Subsection 1.3.6]
$\ \cdot\ $	norm of a random variable [Subsection 3.3.1]
$\perp$	orthogonal vectors [Subsection 1.3.4]
$\perp$	orthogonal random variables [Subsection 3.3.1]
$f(x) _{x=0}$	function evaluated at $x = 0$
$[a, b]$	closed interval between points $a$ and $b$ (includes the endpoints)
$(a, b)$	open interval between points $a$ and $b$ (without the endpoints)
$[a, b)$	semi-open (semi-closed) interval between points $a$ and $b$
$[a_{ij}]$	a matrix whose component at $i$ -row $j$ -column is $a_{ij}$
$x[t_0, t_1]$	[Subsection 1.3.1] the function $x(t)$ in the corresponding interval
$\{x(j)\}_{j=j_0}^{j_1}$	[Subsection 4.2.1] the sequence $x(j)$ , $j = j_0, \dots, j_1$ [Subsection 4.3.3]
$\{x(j), j = j_0, \dots, j_1\}$	the set $\{x(j_0), x(j_0 + 1), \dots, x(j_1)\}$
$\langle \cdot, \cdot \rangle$	inner product of vectors [Subsection 1.3.2]
$\langle \cdot, \cdot \rangle$	inner product of random variables [Subsection 3.3.1]
$\wedge$	the smallest of two variables [Subsection 10.2.4]
$\vee$	the largest of two variables [Subsection 10.2.4]
$\gg$	much larger than
$\triangleq$	equal by definition
$\equiv$	identically equal
$\approx$	approximately equal
$\sim$	distributed as [Subsection 1.4.13]
$\implies$	implies
$\iff$	if and only if — implies and is implied by
$\exists$	there exists
$\ni$	such that
$\in$	element of
$\cup$	logical “OR” operation (set union) [Subsection 1.4.1]
$\cap$	logical “AND” operation (set intersection) [Subsection 1.4.1]
$\nabla_x$	gradient with respect to (the vector) $x$



# Chapter 1

## INTRODUCTION

### 1.1 BACKGROUND

#### 1.1.1 Estimation and Related Areas

Estimation is the process of inferring the value of a quantity of interest from indirect, inaccurate and uncertain observations.

The purpose of estimation can be:

- Determination of planet orbit parameters — probably the first estimation problem — studied by Laplace, Legendre, and Gauss
- Statistical inference
- Determination of the position and velocity of an aircraft in an air traffic control system — tracking
- Application of control to a plant in the presence of uncertainty (noise and/or unknown parameters) — parameter identification, state estimation and stochastic control
- Determination of model parameters for predicting the state of a physical system or forecasting economic or other variables — system identification
- Determination of the characteristics of a transmitted message from noise-corrupted observation of the received signal — communication theory
- Determination of some parameters or characteristics of a signal or image — signal/image processing

More rigorously, **estimation** can be viewed as the *process of selecting a point from a continuous space* — the “best estimate.”

**Decision** can be viewed as the *selection of one out of a set of discrete alternatives* — the “best choice” from a discrete space. However, one can talk about estimation in a discrete-valued case with the possibility of not making a choice but obtaining some conditional probabilities of the various alternatives. This information can be used without making “hard decisions.”

Estimation and decision can, therefore, be seen to be overlapping and techniques from both areas are used simultaneously in many practical problems.

**Tracking** is the estimation of the state of a moving object based on *remote measurements*. This is done using one or more sensors at fixed locations or on moving platforms.

At first sight, tracking might seem to be a special case of estimation. However, it is wider in scope: Not only does it use all the tools from estimation, but it also requires extensive use of statistical decision theory when some of the practical problems (data association — “Which is my measurement?” — see, e.g., [Bar-Shalom95]) are considered.

**Filtering** is the estimation of the (current) state of a dynamic system — the reason for the use of the word “filter” is that the process for obtaining the “best estimate” from noisy data amounts to “filtering out” the noise. The term filtering is thus used in the sense of eliminating an undesired signal, which, in this case, is the noise.

In control systems, signals are also filtered to obtain the estimate of the state of the (noisy) dynamic system, needed by the controller. Filtering of signals is very commonly used in signal processing — in the frequency domain as well as in the spatial domain. The latter is done, for example, to select signals coming from a certain direction. **Navigation** is the estimation of the state of the platform (“own ship”) on which the sensor is located.

An **optimal estimator** is a computational algorithm that processes observations (measurements) to yield an estimate of a variable of interest, which optimizes a certain criterion.

In view of the above discussion, estimation can be viewed as a scheme for **information extraction and enhancement**: Based on measurements (observations), we want to maximize our knowledge about a parameter, a state, a signal, an image, and so on.

In general, one can classify the variable that is to be estimated into the following two categories:

- A parameter — a time-invariant quantity (a scalar, a vector, or a matrix)
- The state of a dynamic system (usually a vector), which evolves in time according to a stochastic equation

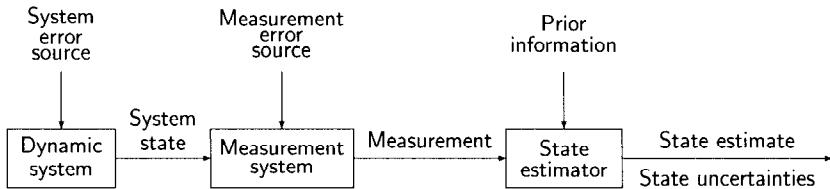
Consequently, one has two classes of estimators:

- Parameter estimators
- State estimators

While some engineering texts deal mainly (or exclusively) with state estimation, which is more general than parameter estimation, a balanced approach is taken here. The reason it is important to understand both is that some results — e.g., convergence and, more importantly, **information limits** — are more readily available/usable for parameter estimators than for state estimators. This

understanding can be helpful in choosing the type of mathematical model used for solving practical problems.

Figure 1.1.1-1 presents a concise block diagram that illustrates state estimation. In this figure the first two blocks are “black boxes” — there is no access to variables inside them. The only variables to which the estimator has access are the **measurements**, which are affected by the error sources in the form of “measurement noise.”



**Figure 1.1.1-1:** State estimation — information extraction and enhancement.

The estimator uses knowledge about the following:

- The evolution of the variable (the system dynamics)
- The measurement model
- The probabilistic characterization of the various random factors (disturbances)
- The prior information

## Optimal Estimation — Advantages and Disadvantages

The advantage of an optimal estimator is that it makes the best utilization of the data and the knowledge of the system and the disturbances.

The disadvantages, like for any optimal technique, are that it is possibly sensitive to modeling errors and might be computationally expensive. In view of this, it is very important to have a clear understanding of the assumptions under which an algorithm is optimal and how they relate to the real world.

## 1.1.2 Applications of Estimation

State estimation, in combination with decision theory, has a great variety of applications. A partial list of the areas where this has found use is as follows:

- Tracking/surveillance and trajectory determination
- Navigation
- Control systems
  - guidance
  - attitude control

- sensor pointing
- steel making
- chemical, nuclear, and industrial processes, etc.
- Power systems
- Failure detection
- Signal processing
- Image processing
- Communications
- Biomedical engineering
- Operations research
- Mapping via remote sensing
- Geophysical problems
- Fluid flow rate measurement
- Econometric systems
  - macroeconomic models
  - microeconomic models
- Demographic systems

### 1.1.3 Preview of Estimation/Filtering

This subsection will present a bird's-eye view of estimation, from the the Least Squares technique originated by Gauss and Legendre for parameter estimation in astronomical studies (motion of heavenly bodies) to Kalman filtering for state estimation. All this will be done without mathematics, to illustrate the concepts.<sup>1</sup>

Estimation is the process of inferring the values of some parameters that characterize, e.g., the motion of a heavenly body (the six parameters of planetary motion — 3 position and 3 velocity — evolving in a predictable manner) or the state of an object, e.g., an aircraft (evolving in a not completely predictable manner).

Gauss made the following (philosophical) observations on the (physical) observations that could be made on the planetary motion:

- If the observations were absolutely correct, the parameters could be determined with perfect accuracy from a minimum number of observations ( $n$  observations for  $n$  parameters).
- Then subsequent observations would confirm, but not correct, the values obtained for the parameters.

---

<sup>1</sup>The alternative title for this subsection could have been "Kalman Filtering for Dummies." This material could be used by readers eager to show their spouses how they make a living; however, we assume no responsibility for the possible ensuing marital problems.

- But, since the observations are only approximations of the truth, we should combine more observations than the minimum to determine more accurately the unknown quantities.
- Thus, starting with approximate knowledge, we can correct it with subsequent observations so as to satisfy all the observations in the most accurate manner possible.

The above (philosophical) observations lead to the following implications:

- A basic description (model) of the “system” is available with some unknown parameters to be estimated.
- Redundant data are required to reduce the effect of measurement errors.
- To satisfy all the observations in the most accurate manner possible, the “residuals” (differences between the observed values and the values predicted from the estimates) should be as small as possible.
- Inaccuracy of observations necessitates probabilistic modeling.
- Combination of the initial knowledge and the subsequent observations leads to the recursive algorithm concept.

Estimation theory provides a systematic approach to deal with the discrepancies in different measurements on the same object.

There are a number of estimation problems that will be briefly outlined below.

### **The Least Squares (Gauss-Legendre) Problem**

If the observations consist of

- linear combinations of the parameters of interest, in the presence of additive errors (noise) assumed to be zero-mean Gaussian random variables,

then the

- minimization of the sum of the squares of the residuals (errors) in fitting the data namely, the least squares (LS) of the difference between the model and the observations,

is equivalent to the

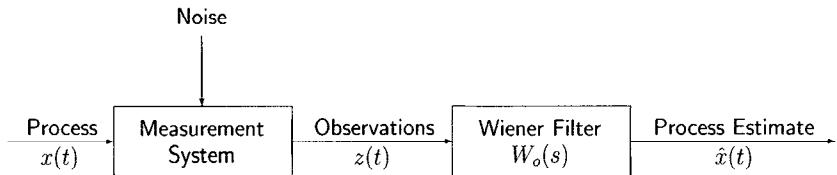
- maximization of the likelihood function of the parameters (the probability density function of the measurements conditioned on the unknown parameters).

## The Wiener-Hopf Problem

The variables to be estimated are not constant but time-varying: The objective is to estimate a random process.

The process to be estimated,  $x(t)$ , is assumed stationary (typically, zero-mean) and with known autocorrelation. The cross-correlation between this process and the observations  $z(t)$  is assumed also known.

The solution that yields the estimate  $\hat{x}(t)$  of the process, obtained in the frequency domain as a transfer function  $W_o(s)$  via Fourier transforms, so as to minimize a mean square error is the Wiener filter. This is illustrated in Fig. 1.1.3-1.



**Figure 1.1.3-1:** The Wiener filter — linear time-invariant system with transfer function  $W_o(s)$ .

## The Kalman-Bucy Problem

The object of the estimation is a random process, which is not necessarily stationary. Its characterization is given by a linear differential (or difference) equation with known coefficients — the state equation of a dynamic system. Random disturbances affect the evolution of the process (process noise). The process is observed in the presence of measurement errors (measurement noise).

The solution obtained (in the time domain) so as to minimize a mean square error is the Kalman-Bucy or Kalman filter.

## Solution to the Least Squares Estimation Problem

The standard (batch) solution to the Least Squares Estimation (LSE) problem consists of the following:

- Given  $k$  measurements (each an  $m$ -vector) of a constant parameter (an  $n$ -vector), the LS estimate of the unknown parameter is obtained by processing simultaneously the entire measurement set (a  $km$ -vector).
- To obtain the new estimate of the parameter when one gets a new measurement (the  $(k+1)$ th  $m$ -vector), one reprocesses the entire data, a  $(k+1)m$ -vector, i.e., a new problem of *larger size* is solved.

The batch LSE solution can be rewritten (if the noise components in the  $k$  sets of measurements are uncorrelated) as a *recursion*

$$(\text{estimate})_{k+1} = (\text{estimate})_k + (\text{weighting})_{k+1} \times (\text{residual})_{k+1}$$

where  $k$  is the time index,  $(\cdot)_k$  indicates the value at time  $k$  and

$$\begin{aligned} (\text{residual})_{k+1} &= (\text{measurement})_{k+1} \\ &\quad - (\text{predicted measurement})_{k+1} \text{ based on } (\text{estimate})_k \end{aligned}$$

The size of this problem is *the same* for every  $k$ .

The accuracy of the estimate also obeys a recursion; that is, the “precision” of the estimate (the inverse of the parameter mean square error — MSE) evolves according to

$$\begin{aligned} (\text{precision})_{k+1} &= (\text{precision})_k \\ &\quad + (\text{information about the parameter in measurement})_{k+1} \end{aligned}$$

Thus the “precision” increases (the MSE or variance decreases) strictly if the measurements contain information about the parameters — if the parameters are *observable*.

If the parameters are constant, as  $k \rightarrow \infty$  one accumulates an unlimited amount of information and the error in the estimates decreases to zero — the LSE converges to the true value. As the MSE in the estimate decreases, so does the weighting in the recursion: as the confidence in the estimates grows, new data are weighted less and less.

## Estimation of the State of a Linear Stochastic Dynamic System

A realistic model for the state of a system has typically

- an imperfect initial estimate
- an imperfectly predictable evolution (a stochastic disturbance affects its dynamics — the process noise)

Similarly, the measurement model is as follows:

- Only some state components (or combinations of them) are observed.
- These observations are inaccurate — corrupted by measurement noise.

The evolution model for the state (discrete time dynamic equation) is as follows:

$$(\text{state})_{k+1} = (\text{linear function of the state})_k + (\text{process noise})_k$$

where  $k$  is the time index and  $(\cdot)_k$  indicates the value at time  $k$ .

The observation process (measurement equation) is

$$(\text{measurement})_k = (\text{linear function of the state})_k + (\text{measurement noise})_k$$

The two noise sequences are assumed to have known means (typically zero) and variances (covariance matrices in the multidimensional case), uncorrelated in time and from each other.

The (discrete time) Kalman filter (KF) computes the best estimate of the current state (in the minimum mean square error sense) based on the measurements up to the current time, denoted as (state estimate)<sub>*k*</sub>. The KF recursion is as follows:

$$(\text{state estimate})_{k+1} = (\text{predicted state})_{k+1^-} + (\text{weighting})_{k+1} \times (\text{residual})_{k+1}$$

where  $k + 1^-$  denotes “just before time  $k + 1$ ” and

$$(\text{residual})_{k+1} = (\text{measurement})_{k+1} - (\text{predicted measurement})_{k+1^-}$$

The LSE recursion is remarkably similar:

$$\begin{aligned} (\text{parameter estimate})_{k+1} &= (\text{parameter estimate})_k \\ &\quad + (\text{weighting})_{k+1} \times (\text{residual})_{k+1} \end{aligned}$$

The accuracy of the estimate (precision — inverse of the variance) is as follows:

$$\begin{aligned} (\text{precision})_{k+1} &= (\text{precision})_{k+1^-} \\ &\quad + (\text{information about the state in measurement})_{k+1} \end{aligned}$$

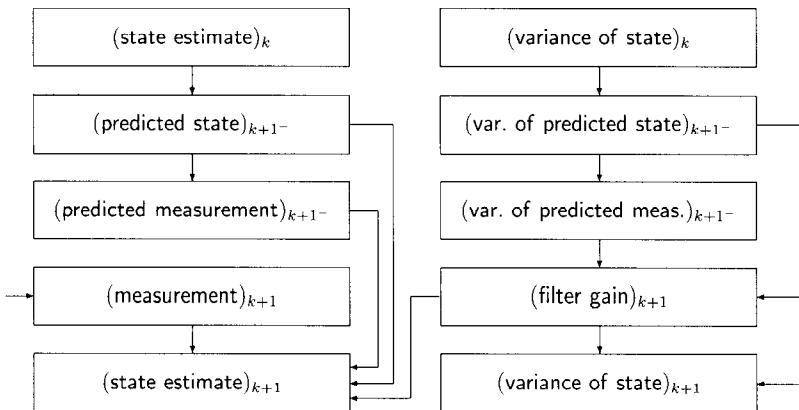
The filter gain reflects the relative accuracy of the predicted state vs. the new observation. The new (updated) state estimate is the optimal combination of

- the entire past data (with the predicted state being the sufficient statistic) and
- the latest measurement

The sequence of the operations in the KF is outlined in Fig. 1.1.3-2. The calculations involving the state estimate use the measurements and have to be done in real time. However, the calculations involving the variances (covariance matrices in the multidimensional case) are data-independent and can be done off-line (before the data are obtained). The variance of the estimates (their “precision”) can be precomputed using the statistical description (variances) of the noise components entering into the system and the initial state variance.

### Illustration for Tracking a Moving Point

Figure 1.1.3-3 illustrates graphically the state estimation process for a point moving with a nearly constant velocity along the *x* axis. The state vector consists of position *x* and velocity  $\dot{x}$ , with estimates  $\hat{x}$  and  $\hat{\dot{x}}$ , respectively. The



**Figure 1.1.3-2:** Sequence of operations in the KF.

covariance matrix of the state is denoted by  $P$  and it indicates the uncertainty region around the estimated state where the true state lies with high probability.

The measurement is

$$z_{k+1} = x_{k+1} + (\text{measurement noise})_{k+1}$$

The new position estimate will be between the observed and predicted values:

$$z_{k+1} < \hat{x}_{k+1} < \hat{x}_{k+1-}$$

Assuming that at time  $k + 1$  the measured position is behind the predicted position, that is,

$$z_{k+1} - \hat{x}_{k+1-} < 0$$

the new velocity estimate will be less than the predicted one:

$$\hat{x}_{k+1} < \hat{x}_{k+1-}$$

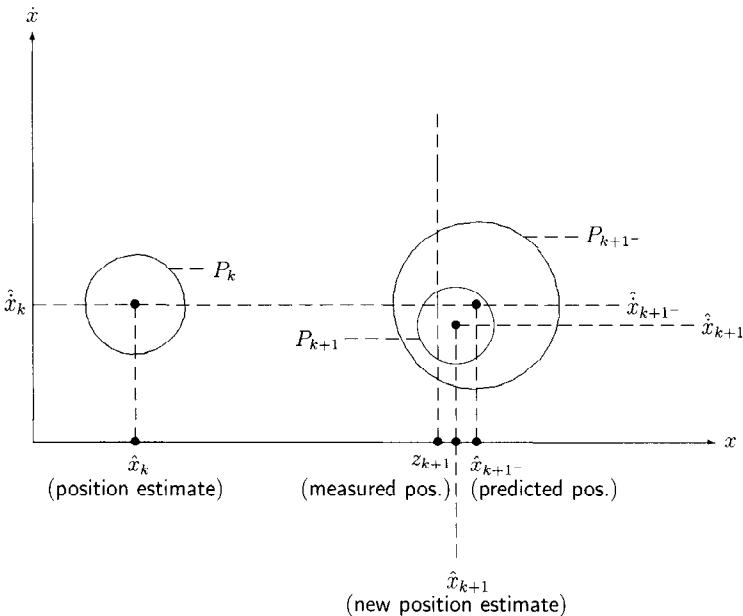
Under stationarity (and state observability) conditions, the variance of the estimates converges to a steady-state value, as illustrated in Fig. 1.1.3-4, where the prediction and updated variances are shown.

## Remarks

The KF is a stochastic counterpart of the (deterministic) state observer.

When the covariance is in steady state, the KF gain becomes a constant — such filters (for “nearly constant velocity” motion models) are known as  $\alpha$ - $\beta$  filters.

The Wiener filter is the frequency domain representation of the steady-state KF.



**Figure 1.1.3-3:** Illustration of the state estimation process.

The sampling interval (time between successive measurements) does not have to be constant in the KF.

Nonstationary disturbances (e.g., measurement noise with time-varying intensity) can be accommodated easily by the KF algorithm.

#### 1.1.4 An Example of State Estimation: Vehicle Collision Avoidance

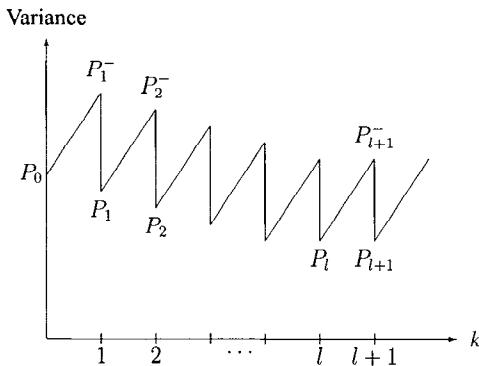
This subsection introduces the problem of state estimation for a dynamic system by considering the estimation of distance and speed between two vehicles as it arises in a collision avoidance system.

An algorithm is presented that estimates the distance (range) between two vehicles and their relative speed (range rate), based on imperfect (noisy) measurements of the distance between them. This algorithm is a simplified version of the Kalman filter, known as the  $\alpha\text{-}\beta$  filter. It can also be viewed as a state observer [Chen84], which, based on observations of one state component, reconstructs (estimates) the entire state vector.

Define the state vector consisting of range  $r$  and range rate  $\dot{r}$  as

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} r \\ \dot{r} \end{bmatrix} \quad (1.1.4-1)$$

Assuming a constant speed (constant range-rate), the state equation is, for



**Figure 1.1.3-4:** Evolution of the variance of the estimate with time.

discrete time with sampling period  $T$ ,

$$\begin{bmatrix} r(k) \\ \dot{r}(k) \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r(k-1) \\ \dot{r}(k-1) \end{bmatrix} = F \begin{bmatrix} r(k-1) \\ \dot{r}(k-1) \end{bmatrix} \quad (1.1.4-2)$$

where the system matrix is

$$F = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad (1.1.4-3)$$

The measurement is

$$y(k) = x_1(k) = r(k) - [1 \ 0]x(k) = Hx(k) \quad (1.1.4-4)$$

where the measurement matrix is

$$H = [1 \ 0] \quad (1.1.4-5)$$

In truth, the measurement is a “corrupted” (noisy) version of (1.1.4-4), i.e.,

$$y(k) = Hx(k) + w(k) \quad (1.1.4-6)$$

where  $w(k)$  is the (range) measurement error.

Similarly, the state equation should reflect an acceleration, denoted as  $u$ , namely,

$$x(k) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x(k-1) + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} u(k-1) \quad (1.1.4-7)$$

Using (1.1.4-2) instead of (1.1.4-7), since we do not know  $u$ , as the “motion model” for the estimation problem leads to motion modeling errors.

## The Estimator

In view of the fact that there are both measurement errors and motion modeling errors, we shall calculate the “estimate”  $\hat{x}(k)$  of the state vector  $x(k)$  via the

following algorithm, which is a simplified form of the Kalman filter, known as the  $\alpha\text{-}\beta$  filter (see Chapters 5 and 6).

$$\hat{x}(k) = \bar{x}(k) + W[y(k) - \bar{y}(k)] \quad (1.1.4-8)$$

where  $\bar{x}(k)$  is the “predicted state” at time  $k$  from  $k-1$

$$\begin{aligned} \bar{x}(k) &= F\hat{x}(k-1) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \hat{x}(k-1) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{r}(k-1) \\ \hat{\dot{r}}(k-1) \end{bmatrix} \\ &= \begin{bmatrix} \hat{r}(k-1) + T\hat{\dot{r}}(k-1) \\ \hat{\dot{r}}(k-1) \end{bmatrix} \end{aligned} \quad (1.1.4-9)$$

and  $\bar{y}(k)$  is the “predicted measurement” at  $k$  from  $k-1$

$$\bar{y}(k) = H\bar{x}(k) = [1 \ 0]\bar{x}(k) = \hat{r}(k-1) + T\hat{\dot{r}}(k-1) \quad (1.1.4-10)$$

The difference between the measurement (observed range) and its predicted value

$$y(k) - \bar{y}(k) = y(k) - H\bar{x}(k) \quad (1.1.4-11)$$

is the “residual,” which is multiplied (weighted) by the “filter gain”

$$W = \begin{bmatrix} \alpha \\ \beta/T \end{bmatrix} \quad (1.1.4-12)$$

The residual multiplied by the gain is used in (1.1.4-8) to “correct” or “update” the predicted state to obtain the new estimate  $\hat{x}(k)$ . The filter (1.1.4-8) can be seen as a ***predictor-corrector algorithm***.

The (dimensionless) constants  $\alpha$  and  $\beta$  are such that  $0 < \alpha < 1$  and  $0 < \beta < 2$ . The filter gain coefficients  $\alpha$  and  $\beta$  define the  $\alpha\text{-}\beta$  filter (1.1.4-8).

The breakdown of the state estimation equation (1.1.4-8) for its two components  $r$  and  $\dot{r}$  is

$$\hat{r}(k) = \hat{r}(k-1) + T\hat{\dot{r}}(k-1) + \alpha[y(k) - \bar{y}(k)] \quad (1.1.4-13)$$

$$\hat{\dot{r}}(k) = \hat{\dot{r}}(k-1) + \frac{\beta}{T}[y(k) - \bar{y}(k)] \quad (1.1.4-14)$$

The intuitive interpretation of (1.1.4-13) is as follows:

- If the predicted range is larger than the measured one, the residual will be negative and a negative correction is made with gain  $\alpha < 1$  to obtain the new estimated range.
- If the predicted range is smaller than the measured one, the residual will be positive, and a positive correction will be made.

Similarly, the range rate (1.1.4-14) is corrected downward for a negative residual and upward for a positive one. Also note that the residual has the dimension of length and, when multiplied by  $\beta/T$ , its dimension becomes length/time, i.e., velocity.

## A Deterministic Error Analysis

Assuming a constant acceleration (or deceleration)  $u$ , the state equation is, based on (1.1.4-7), given by

$$x(k) = Fx(k-1) + Gu \quad (1.1.4-15)$$

where

$$G = \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \quad (1.1.4-16)$$

reflects the effect of the acceleration  $u$  on the range ( $uT^2/2$ ) and on the range rate ( $uT$ ).

Assuming the measurements to be given by (1.1.4-4), i.e., neglecting measurement errors, the evaluation of the effect of the acceleration on the errors in the range and range-rate estimates is carried out next.

The error is defined as the difference between the actual value and the estimate, namely,

$$\tilde{x}(k) = x(k) - \hat{x}(k) \quad (1.1.4-17)$$

From (1.1.4-8), using (1.1.4-9) and (1.1.4-10), the estimate is given by

$$\hat{x}(k) = F\hat{x}(k-1) + W[y(k) - HF\hat{x}(k-1)] \quad (1.1.4-18)$$

Substituting (1.1.4-4) for  $y(k)$  in the above yields

$$\hat{x}(k) = F\hat{x}(k-1) + WH[x(k) - F\hat{x}(k-1)] \quad (1.1.4-19)$$

Substituting (1.1.4-15) for  $x(k)$  in the above yields

$$\hat{x}(k) = F\hat{x}(k-1) + WH[Fx(k-1) + Gu - F\hat{x}(k-1)] \quad (1.1.4-20)$$

or, using notation (1.1.4-17),

$$\hat{x}(k) = F\hat{x}(k-1) + WHF\tilde{x}(k-1) + WHGu \quad (1.1.4-21)$$

Subtracting (1.1.4-21) from (1.1.4-15) yields

$$x(k) - \hat{x}(k) = Fx(k-1) + Gu - F\hat{x}(k-1) - WHF\tilde{x}(k-1) - WHGu \quad (1.1.4-22)$$

which can be rewritten as

$$\tilde{x}(k) = [I - WH]F\tilde{x}(k-1) + [I - WH]Gu \quad (1.1.4-23)$$

Note that the dynamic equation obeyed by the error  $\tilde{x}$  is a state equation with system matrix

$$\tilde{F} = [I - WH]F = \begin{bmatrix} 1 - \alpha & (1 - \alpha)T \\ -\frac{\beta}{T} & 1 - \beta \end{bmatrix} \quad (1.1.4-24)$$

and input with gain

$$\tilde{G} = [I - WH]G = \begin{bmatrix} (1-\alpha)T^2/2 \\ (1-\frac{\beta}{2})T \end{bmatrix} \quad (1.1.4-25)$$

i.e.,

$$\tilde{x}(k) = \tilde{F}\tilde{x}(k-1) + \tilde{G}u \quad (1.1.4-26)$$

It can be shown that the inequalities indicated earlier for  $\alpha$  and  $\beta$  guarantee that the matrix  $\tilde{F}$  above is stable, i.e., it has eigenvalues inside the unit circle. In other words, for a constant “disturbance”  $u$ , the error  $\tilde{x}(k)$  converges to a steady-state value  $\tilde{x}_{ss}$ , which is given by the expression

$$\tilde{x}_{ss} = (I - \tilde{F})^{-1}\tilde{G}u = \begin{bmatrix} \frac{1-\alpha}{\beta}T^2 \\ \frac{2\alpha-\beta}{2\beta}T \end{bmatrix} u \quad (1.1.4-27)$$

For  $u = 5 \text{ m/s}^2$ ,  $T = 0.1 \text{ s}$ ,  $\alpha = 0.75$ ,  $\beta = 0.5$ , one obtains  $\tilde{r}_{ss} = 0.39 \text{ m}$ ,  $\dot{\tilde{r}}_{ss} = 0.5 \text{ m/s}$ .

### Frequency Domain View of Estimation: The Bandwidth of a Filter

Based on (1.1.4-23), one has the following  $z$ -domain transfer function from the input to the error  $\tilde{x}$ :

$$\tilde{H}(z) = \frac{\tilde{X}(z)}{U(z)} = (zI - \tilde{F})^{-1}\tilde{G} \quad (1.1.4-28)$$

With the above numbers,

$$\tilde{F} = \begin{bmatrix} 0.25 & 0.025 \\ -5 & 0.5 \end{bmatrix} \quad \tilde{G} = \begin{bmatrix} 0.005 \\ 0.1 \end{bmatrix} \quad (1.1.4-29)$$

the transfer function from the input to the position component error is

$$\tilde{H}_1(z) = \frac{\tilde{X}_1(z)}{U(z)} = \frac{0.005z}{z^2 - 0.75z + 0.25} \quad (1.1.4-30)$$

The two poles of  $\tilde{H}_1(z)$  have magnitude  $1/2$ , i.e., this is a stable transfer function.

Figure 1.1.4-1 shows the plot of the magnitude of the above transfer function for the range of frequencies  $[0, \frac{1}{2T}]$  (the upper limit is the point for which the Nyquist frequency is the sampling frequency). It can be clearly seen that this is a low-pass filter with a bandwidth (based on the half-power point) of approximately 2 Hz.

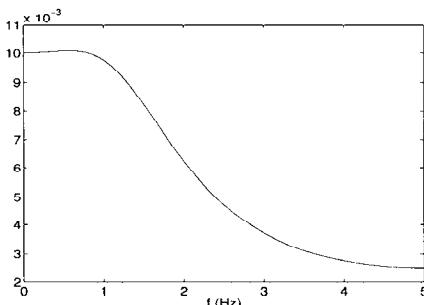


Figure 1.1.4-1: The frequency response of the state estimation filter.

## 1.2 SCOPE OF THE TEXT

### 1.2.1 Objectives

The objectives of this text are to present the fundamentals of *state estimation theory* and the tools for the design of state-of-the-art algorithms for *target tracking and navigation*.

The text covers the basic concepts and estimation techniques for static and dynamic systems, linear and nonlinear, as well as adaptive estimation. This constitutes a one-semester graduate course in estimation theory in an electrical/systems engineering program.

Special emphasis is given to the statistical tools that can be used for the *interpretation of the output from stochastic systems*. These are key tools for the assessment of the performance of state estimation/tracking filters — in simulations as well as in real-time implementation.

The discussion deals mainly with discrete time estimation algorithms, which are natural for digital computer implementation. Also, typically, sensors provide observations in discrete time. The basic state estimation algorithm — the Kalman filter — is presented in discrete as well as in continuous time.

The use of the estimation algorithms is illustrated on kinematic motion models because they reveal all the major issues and, in particular, the subtleties encountered in estimation. Also this serves as an introduction to tracking.

Guidelines for the *design of tracking filters*, namely,

- selection of the filter design parameters
- selection of the type of filter — nonadaptive (single model based) vs. adaptive (multiple model based)

are given and illustrated in several examples. Most of the examples are run with the companion software DynaEst<sup>TM</sup>; also, a step-by-step demonstration of how to use DynaEst<sup>TM</sup> is provided.

## Prerequisite Material

The *linear algebra tools* that form the backbone of the state space analysis techniques used in estimation/tracking are presented in Section 1.3 together with some useful results from linear systems. Matrix notation will be used throughout this text — every quantity should be considered a matrix, unless otherwise stated.

A review of the *tools from probability theory and random processes*<sup>2</sup> is presented in Section 1.4.

Section 1.5 reviews *statistical hypothesis testing* and presents some useful statistical tables for the chi-square and the normal (Gaussian) distributions. Particular attention is given to the concept of statistical significance, which, while common in the statistics literature, has not received much attention in the engineering literature.

These review sections cover all the background material needed for the text and the basic concepts are illustrated by examples. This might be used as a (crash) minicourse to bring advanced undergraduate students up to the level necessary to understand the main material of the text.

### 1.2.2 Overview and Chapter Prerequisites

In the following an overview of the text is given with the prerequisite to each chapter.

#### Review of Background Techniques (remainder of Chapter 1)

- Linear algebra
- Probability theory and stochastic processes
- Statistics

#### Basic Concepts in Estimation — Chapter 2 (prerequisite: 1)

- Estimation techniques (ML, MAP, LS and MMSE)
- Properties of estimators
- Limit of existing information in the data

#### Linear Estimation in Static Systems — Chapter 3 (prerequisite: 2)

- Techniques (LMMSE, recursive LS and prelude to Kalman filtering)
- Application to polynomial fitting

#### Linear Dynamic Systems with Random Inputs — Chapter 4 (prerequisite: 1)

- Models — continuous and discrete time state space models
- Response of dynamic systems to random inputs

---

<sup>2</sup>Probability demonstrates both our knowledge as well as our ignorance.

State Estimation in Discrete-Time Linear Dynamic Systems — Chapter 5 (prerequisites: 3, 4)

- The Kalman filter and its properties
- Use of the companion software DynaEst™ to design and evaluate a KF
- Consistency of state estimators
- Initialization — in simulations and in practice

Estimation for Kinematic Models — Chapter 6 (prerequisite: 5)

- Types of kinematic models
- Explicit filters and the target maneuvering index
- Filter design and system performance prediction
- Sampling interval selection (system design)

Computational Aspects of Estimation — Chapter 7 (prerequisite: 5)

- The Information filter
- Sequential update
- Square-root filtering

Extensions of Discrete-Time Linear Estimation — Chapter 8 (prerequisite: 5)

- Correlated process noise
- Cross-correlated process and measurement noise sequences
- Correlated measurement noise
- Prediction
- Smoothing

Continuous-Time Linear State Estimation — Chapter 9 (prerequisite: 5)

- The Kalman-Bucy filter and its properties
- Prediction
- Duality between estimation and control
- The Wiener-Hopf problem

State Estimation for Nonlinear Dynamic Systems — Chapter 10 (prerequisite: 5)

- Optimal estimation
- Suboptimal estimator — the extended Kalman filter (EKF)
- Practical issues in implementation
- Trajectory estimation via dynamic programming

Adaptive Estimation and Maneuvering Targets — Chapter 11 (prerequisite: 5)

- White and colored noise models
- Input estimation
- Variable state dimension filtering

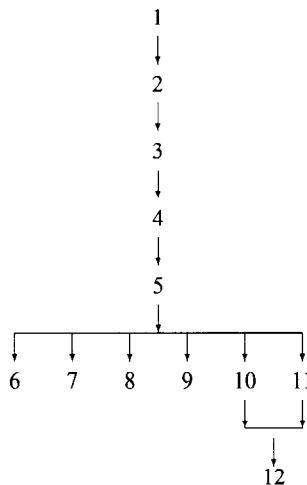
- Comparison of several methods
- The multiple model approach — static and dynamic, the Interacting Multiple Model (IMM) estimator
- Use of the companion software DynaEst<sup>TM</sup> to design and evaluate an IMM estimator
- An air traffic control (ATC) tracking algorithm design example with the IMM estimator
- Estimator selection — nonadaptive vs. adaptive
- Use of the EKF for parameter estimation

Estimation for Navigation Systems — Chapter 12 (prerequisites: 5 and, for the last part, 11)

- Inertial navigation systems (INS)
- Global Positioning System (GPS)
- Combination of INS and GPS
- Use of the KF and IMM estimator for GPS navigation

At the end of each chapter, a number of problems that enhance the understanding of the theory and connect the theoretical material to the real world are given.

The sequence of prerequisites is illustrated in Fig. 1.2.2-1.



**Figure 1.2.2-1:** Sequence of chapter prerequisites.

Two major examples of realistic systems are presented:

- A bearings-only target localization from a moving platform based on the Maximum Likelihood technique
- An ATC problem for tracking maneuvering aircraft using an adaptive estimation technique (the IMM estimator)

As part of the problems at the end of Chapter 3, a project is given to develop an interactive program for the first example listed above.

A major term project that involves a realistic ATC tracking problem (with nonlinear measurements from a radar — range and azimuth) is given at the end of Chapter 11. The sequence of prerequisites for this is Chapters 1–6, Section 10.3 and Section 11.6.

An alternative major term project that involves a realistic ATC navigation problem with GPS is given at the end of Chapter 11. The sequence of prerequisites for this is Chapters 1–6, Section 10.3 and Sections 12.6–12.8.

## 1.3 BRIEF REVIEW OF LINEAR ALGEBRA AND LINEAR SYSTEMS

### 1.3.1 Definitions and Notations

A **matrix** of dimension  $n \times m$  is the two-dimensional array

$$A = [a_{ij}] = \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{bmatrix} \quad (1.3.1-1)$$

The first dimension is the number of rows, and the second dimension is the number of columns. The elements of  $A$  will be denoted as  $a_{ij}$  or as  $A_{ij}$ .

An  $n$ -dimensional **vector** is the one-dimensional array ( $n \times 1$  matrix)

$$a = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} \triangleq \text{col}(a_i) \quad (1.3.1-2)$$

In this text all vectors will be **column vectors**.

**Transposition** of a matrix or vector will be denoted by an apostrophe. The **transpose** of the matrix (1.3.1-1) is

$$A' = [a_{ji}] = \begin{bmatrix} a_{11} & \cdots & a_{n1} \\ \vdots & \ddots & \vdots \\ a_{1m} & \cdots & a_{nm} \end{bmatrix} \quad (1.3.1-3)$$

With this notation one can write the column vector (1.3.1-2) as

$$a = [a_1 \cdots a_n]' \quad (1.3.1-4)$$

The **row vector** obtained from transposing (1.3.1-2) is

$$a' = [a_1 \cdots a_n] \quad (1.3.1-5)$$

A (square) matrix is said to be **symmetric** if

$$A = A' \quad (1.3.1-6)$$

or

$$a_{ij} = a_{ji} \quad \forall i, j \quad (1.3.1-7)$$

### 1.3.2 Some Linear Algebra Operations

Addition of matrices and the multiplication of a matrix by a scalar are defined as follows. With  $\alpha$  and  $\beta$  scalars, the matrix

$$C = \alpha A + \beta B \quad (1.3.2-1)$$

has elements given by

$$c_{ij} = \alpha a_{ij} + \beta b_{ij} \quad i = 1, \dots, n; \quad j = 1, \dots, m \quad (1.3.2-2)$$

where all the matrices have the same dimension  $n \times m$ .

The product of two matrices

$$C = AB \quad (1.3.2-3)$$

has elements

$$c_{ij} = \sum_{k=1}^m a_{ik} b_{kj} \quad i = 1, \dots, n; \quad j = 1, \dots, p \quad (1.3.2-4)$$

where  $A$  is  $n \times m$ ,  $B$  is  $m \times p$ , and the result  $C$  is  $n \times p$ . The matrix product is, in general, not commutative.

The *transpose of a product* is

$$C' = (AB)' = B'A' \quad (1.3.2-5)$$

Thus, if

$$Ab = c \quad (1.3.2-6)$$

where  $A$  is  $n \times m$ ,  $b$  is  $m \times 1$  (i.e., an  $m$ -vector) and  $c$  is  $n \times 1$  ( $n$ -vector), then

$$c' = b'A' \quad (1.3.2-7)$$

where  $c'$  is  $1 \times n$  ( $n$ -vector in row form), etc. The *inner product* of two (real)  $n$ -vectors in a Euclidean space is

$$\langle a, b \rangle \triangleq a'b = \sum_{i=1}^n a_i b_i \quad (1.3.2-8)$$

The inner product has a counterpart in probability theory where it is used for random variables, which can be looked upon as vectors in a suitable space. This has applications in linear minimum mean square error estimation.

The *outer product* of two vectors is the matrix

$$ab' = C = [c_{ij}] \quad \text{with} \quad c_{ij} = a_i b_j \quad (1.3.2-9)$$

The *trace* of an  $n \times n$  matrix is the sum of its diagonal elements

$$\text{tr}(A) \triangleq \sum_{i=1}^n a_{ii} = \text{tr}(A') \quad (1.3.2-10)$$

It can be easily shown<sup>3</sup> that if  $A$  is  $m \times n$  and  $B$  is  $n \times m$ , then

$$\text{tr}(AB) = \text{tr}(BA) \quad (1.3.2-11)$$

For example, if  $a$  and  $b$  are both  $n$ -vectors, then

$$a'b = \text{tr}(a'b) = \text{tr}(ba') \quad (1.3.2-12)$$

The first equality above is due to the fact that the trace of a scalar is itself.

It can be shown<sup>4</sup> that for matrices of suitable dimension,

$$\text{tr}(ABC) = \text{tr}(BCA) = \text{tr}(CAB) \quad (1.3.2-13)$$

that is, the trace is *invariant under circular permutations* in its argument. A (square) matrix is ***idempotent*** if and only if<sup>5</sup> for all positive integers  $n$

$$A^n = A \quad (1.3.2-14)$$

The ***square root*** of the square matrix  $A$  is the (in general, nonunique) matrix  $A^{1/2}$ , such that

$$A^{1/2}A^{1/2} = A \quad (1.3.2-15)$$

A symmetric matrix  $A$  can also be decomposed as

$$A = CC' = LDL' \quad (1.3.2-16)$$

where  $C$  is ***lower triangular*** (with zeros above the diagonal),  $L$  is ***unit lower triangular*** (with zeros above the diagonal and units on the diagonal) and  $D$  is ***diagonal***. The first decomposition above is the ***Cholesky factorization*** whereas the last form above is known as the ***LDL'* factorization**, or, since  $L' = U$  is a unit upper triangular matrix, the ***LDU factorization***.

### 1.3.3 Inversion and the Determinant of a Matrix

The ***inverse***  $A^{-1}$  of the  $n \times n$  matrix  $A$  is such that

$$A^{-1}A = I \quad (1.3.3-1)$$

where

$$I = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \triangleq \text{diag}(1) \quad (1.3.3-2)$$

<sup>3</sup>Also known in mathematical circles as the ICBES argument.

<sup>4</sup>Also known in mathematical circles as the ICBS argument.

<sup>5</sup>The standard shorthand for “if and only if” in mathematical texts is “iff.”

is the **identity matrix** of the same dimension as  $A$ .

The inverse is given by the expression

$$A^{-1} = \frac{1}{|A|} C' \quad (1.3.3-3)$$

where  $C$ , called the **adjugate** of  $A$ , is the matrix of cofactors of  $A$  and  $|A|$  is the **determinant** of  $A$ . The cofactor  $(i, j)$  of  $A$  is the determinant of the matrix  $A$  without its row  $i$  and column  $j$ , multiplied by  $(-1)^{i+j}$ .

The inverse of a (square) matrix exists if and only if its columns  $a_i$ ,  $i = 1, \dots, n$  (or its rows) are **linearly independent**, that is,

$$\sum_{i=1}^n \alpha_i a_i = 0 \iff \alpha_i = 0 \quad i = 1, \dots, n \quad (1.3.3-4)$$

where 0 denotes the zero vector (of dimension  $n$ ). This is equivalent to the determinant in (1.3.3-3) being nonzero.

An invertible matrix is also called **nonsingular**.

The determinant of an  $n \times n$  matrix multiplied by a scalar is

$$|\alpha A| = \alpha^n |A| \quad (1.3.3-5)$$

This is useful in writing the probability density function of Gaussian random vector.

## Inversion of a Partitioned Matrix

The inverse of the (nonsingular)  $n \times n$  **partitioned matrix**

$$\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}^{-1} = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix} \quad (1.3.3-6)$$

where  $P_{11}$  is  $n_1 \times n_1$ ,  $P_{12}$  is  $n_1 \times n_2$ ,  $P_{21}$  is  $n_2 \times n_1$ ,  $P_{22}$  is  $n_2 \times n_2$  and  $n_1 + n_2 = n$ , has the partitions

$$V_{11} = P_{11}^{-1} + P_{11}^{-1} P_{12} V_{22} P_{21} P_{11}^{-1} = (P_{11} - P_{12} P_{22}^{-1} P_{21})^{-1} \quad (1.3.3-7)$$

$$V_{12} = -P_{11}^{-1} P_{12} V_{22} = -V_{11} P_{12} P_{22}^{-1} \quad (1.3.3-8)$$

$$V_{21} = -V_{22} P_{21} P_{11}^{-1} = -P_{22}^{-1} P_{21} V_{11} \quad (1.3.3-9)$$

$$V_{22} = P_{22}^{-1} + P_{22}^{-1} P_{21} V_{11} P_{12} P_{22}^{-1} = (P_{22} - P_{21} P_{11}^{-1})^{-1} P_{12} \quad (1.3.3-10)$$

## The Matrix Inversion Lemma

Another useful result is the following identity known as the **matrix inversion lemma**,

$$(P^{-1} + H'R^{-1}H)^{-1} = P - PH'(HPH' + R)^{-1}HP \quad (1.3.3-11)$$

where  $P$  is  $n \times n$ ,  $H$  is  $m \times n$ , and  $R$  is  $m \times m$ .

An alternative version of the above is

$$(A + BCB')^{-1} = A^{-1} - A^{-1}B(B'A^{-1}B + C^{-1})^{-1}B'A^{-1} \quad (1.3.3-12)$$

It can be shown that (1.3.3-6) to (1.3.3-10) and (1.3.3-11) hold by verifying that the corresponding multiplications will yield the identity matrix.

These results have direct application in the derivation of the recursive form of the least squares estimation of parameters as well as in linear estimation for dynamic systems — they yield various forms of the Kalman filter.

### 1.3.4 Orthogonal Projection of Vectors

The inner product of a vector with itself

$$\langle a, a \rangle \triangleq \|a\|^2 \quad (1.3.4-1)$$

is the *squared  $l_2$  norm* of this vector. This applies for the inner product defined in (1.3.2-8) or for any other properly defined inner product. The  $l_2$  norm is the *magnitude* of the vector or its *length*.

The *Schwarz inequality* states the following relationship between the magnitude of the inner product of two vectors and their norms

$$|\langle a, b \rangle| \leq \|a\| \|b\| \quad (1.3.4-2)$$

Two vectors are *orthogonal*, which is denoted as

$$a \perp b \quad (1.3.4-3)$$

if

$$\langle a, b \rangle = 0 \quad (1.3.4-4)$$

The *orthogonal projection* of the vector  $a$  on  $b$  is

$$\Pi_b(a) = \frac{\langle a, b \rangle}{\|b\|^2} b \quad (1.3.4-5)$$

It can be easily shown that the difference vector between  $a$  and its orthogonal projection on  $b$  is orthogonal on  $b$ , that is,

$$[a - \Pi_b(a)] \perp b \quad (1.3.4-6)$$

Figure 1.3.4-1 illustrates this.

The orthogonal projection is used to obtain the linear minimum mean square error (LMMSE) estimates of random variables.

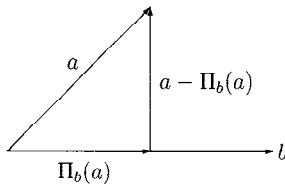


Figure 1.3.4-1: Orthogonal projection of vector  $a$  on vector  $b$ .

### 1.3.5 The Gradient, Jacobian and Hessian

The **gradient** operator with respect to the  $n$ -vector  $x$  is defined as

$$\nabla_x = \left[ \frac{\partial}{\partial x_1} \cdots \frac{\partial}{\partial x_n} \right]' \quad (1.3.5-1)$$

The gradient of an  $m$ -dimensional vector-valued function  $f(x)$  is

$$\nabla_x f(x)' = \begin{bmatrix} \frac{\partial}{\partial x_1} \\ \vdots \\ \frac{\partial}{\partial x_n} \end{bmatrix} [f_1(x) \cdots f_m(x)] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_1}{\partial x_n} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \quad (1.3.5-2)$$

The transpose of the above is the **Jacobian matrix**, an  $m \times n$  matrix in this case

$$J_x(x) \triangleq \frac{\partial f}{\partial x} = [\nabla_x f(x)']' = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \quad (1.3.5-3)$$

The dimensions  $m$  and  $n$  are usually (but not necessarily) equal.

It can be easily shown that

$$\nabla_x x' = I \quad (1.3.5-4)$$

and, if  $A$  is a symmetric matrix,

$$\nabla_x(x'Ax) = 2Ax \quad (1.3.5-5)$$

The **Hessian** of the scalar function  $\phi(x)$  with respect to the  $n$ -vector  $x$  is

$$\phi_{xx}(x) \triangleq \frac{\partial^2 \phi(x)}{\partial x^2} = \nabla_x \nabla'_x \phi(x) = \begin{bmatrix} \frac{\partial^2 \phi}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 \phi}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 \phi}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 \phi}{\partial x_n \partial x_n} \end{bmatrix} \quad (1.3.5-6)$$

which is, obviously, a symmetric  $n \times n$  matrix.

These results are used in the linear estimation of the state of nonlinear systems via series expansion.

### 1.3.6 Eigenvalues, Eigenvectors, and Quadratic Forms

#### Eigenvalues and Eigenvectors

The **eigenvalues** of an  $n \times n$  matrix  $A$  are the scalars  $\lambda_i$  such that

$$Au_i = \lambda_i u_i \quad i = 1, \dots, n \quad (1.3.6-1)$$

where the vectors  $u_i$  are the corresponding **eigenvectors**. If the eigenvalues are distinct, then there are  $n$  eigenvectors; otherwise the number of the linearly independent eigenvectors might be less than  $n$ .

A matrix is **nonsingular** if and only if all its eigenvalues are nonzero. This follows from the fact that its determinant is the product of its eigenvalues

$$|A| = \prod_{i=1}^n \lambda_i \quad (1.3.6-2)$$

The **rank** of a (not necessarily square) matrix is equal to the number of its linearly independent rows or columns. For a square ( $n \times n$ ) matrix, this equals the number of its nonzero eigenvalues.

A nonsingular matrix must have **full rank**. In the case of a full rank  $n \times n$  matrix, its columns (and rows) span the  $n$ -dimensional space.

The eigenvalues of a real matrix can be real or complex; however, a *symmetric* real matrix has only real eigenvalues.

It can be shown that the trace of a matrix equals the sum of its eigenvalues

$$\text{tr}(A) = \sum_{i=1}^n \lambda_i \quad (1.3.6-3)$$

An **idempotent** matrix, defined in (1.3.2-14), that is, with the property

$$AA = A \quad (1.3.6-4)$$

has eigenvalues that are *either zero or unity*.

#### Quadratic Forms

The (scalar) function of the real vector  $x$

$$q = x'Ax \quad (1.3.6-5)$$

is called a **quadratic form**. It can be easily shown that, without loss of generality, in a quadratic form the matrix  $A$  can be considered symmetric.

A **positive (semi)definite quadratic form** is one which is positive (nonnegative) for all nonzero vectors  $x$ .

A **positive (semi)definite matrix** is one for which the quadratic form (1.3.6-5) is positive (semi)definite. This can be summarized as follows:

$$A > 0 \iff x'Ax > 0 \quad \forall x \neq 0 \quad (1.3.6-6)$$

$$A \geq 0 \iff x' A x \geq 0 \quad \forall x \neq 0 \quad (1.3.6-7)$$

If the matrix  $A$  is positive definite, the expression on the right hand side of (1.3.6-5) can also be called the **squared norm with respect to  $A$**  of the vector  $x$  and denoted sometimes as  $\|x\|_A^2$ .

A matrix is positive (semi)definite if and only if all its eigenvalues are positive (nonnegative).

The **inequality of two matrices** is defined as follows: the matrix  $A$  is smaller (not larger) than the matrix  $B$  if and only if the difference  $B - A$  is positive (semi)definite. The outer product of a vector with itself,  $aa'$ , called a **dyad**, is a positive semidefinite matrix (it has rank one).

The sum of dyads of  $n$ -vectors  $a_i$  with positive weights

$$\tilde{A} = \sum_{i=1}^m \alpha_i a_i a_i' \quad \alpha_i > 0 \quad (1.3.6-8)$$

is (at least) positive semidefinite. If the vectors  $a_1, \dots, a_m$  span the  $n$ -dimensional space, in which case one needs  $m \geq n$ , then (1.3.6-8) is positive definite. The **spectral representation** of an  $n \times n$  matrix that has  $n$  eigenvectors is

$$A = \sum_{i=1}^n \lambda_i u_i v_i' \quad (1.3.6-9)$$

where  $\lambda_i$  are its eigenvalues,  $u_i$  are its eigenvectors, and  $v_i$ ,  $i = 1, \dots, n$ , is the **reciprocal basis**, that is,

$$u_i' v_j = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \quad (1.3.6-10)$$

The vectors  $v_i$  are the columns of the inverse of the matrix consisting of the eigenvectors  $u_i'$  as rows.

If the  $n \times n$  real matrix  $A$  is *symmetric* with nonzero eigenvalues, it will always have  $n$  eigenvectors that are *orthogonal*. Furthermore, in this case,

$$v_i = u_i \quad (1.3.6-11)$$

Thus, the spectral representation of a *symmetric* real  $n \times n$  matrix  $A$  is

$$A = \sum_{i=1}^n \lambda_i u_i u_i' \quad (1.3.6-12)$$

The **condition number** of a positive definite symmetric matrix is (usually taken as) the common logarithm of the ratio of its largest to its smallest eigenvalue:

$$\kappa(A) \triangleq \log_{10} \frac{\lambda_{\max}}{\lambda_{\min}} \quad (1.3.6-13)$$

Large condition numbers indicate near-singularity (e.g.,  $\kappa > 6$  for a 32-bit computer indicates an ill-conditioned matrix).

The (induced) **norm of a matrix** is

$$\|A\| \triangleq \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} \quad (1.3.6-14)$$

A consequence of the above and the Schwarz inequality is

$$\|Ax\| \leq \|A\| \|x\| \quad (1.3.6-15)$$

Also

$$\|AB\| \leq \|A\| \|B\| \quad (1.3.6-16)$$

### 1.3.7 Continuous-Time Linear Dynamic Systems — Controllability and Observability

#### The State-Space Representation

The state space representation of **continuous-time linear systems** is

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \quad (1.3.7-1)$$

where

$x(t)$  is the state vector of dimension  $n_x$ ,

$u(t)$  is the input (control) vector of dimension  $n_u$ ,

$A(t)$  is an  $n_x \times n_x$  matrix, called the **system matrix**,

$B(t)$  is an  $n_x \times n_u$  matrix, called the (continuous-time) **input gain**.

Equation (1.3.7-1) is known as the **dynamic equation** or the **plant equation**.

The output of the system is, in general, a vector

$$z(t) = C(t)x(t) \quad (1.3.7-2)$$

of dimension  $n_z$  and  $C$  a known  $n_z \times n_x$  matrix, called the **measurement matrix**.

Equation (1.3.7-2) is known as the **output equation** or the **measurement equation**.

Given the initial condition  $x(t_0)$  and the input function denoted as

$$u_{[t_0,t]} = \{u(\tau), t_0 \leq \tau \leq t\} \quad (1.3.7-3)$$

one can compute the future output at any time  $t > t_0$

$$z(t) = z[x(t_0), u_{[t_0,t]}, t, t_0] \quad (1.3.7-4)$$

The **state** (of a deterministic system) is defined as the smallest dimension vector that *completely summarizes the past of the system*.

A continuous-time system is **linear time-invariant** if it is described by (1.3.7-1) and (1.3.7-2) with  $A(t) = A$ ,  $B(t) = B$ , and  $C(t) = C$ .

## Controllability

A continuous-time (deterministic) system is **completely controllable** if, given an *arbitrary destination point* in the state space, there is an input function that will bring the system from any initial state to this point in a finite time.

For a *linear time-invariant system*, the controllability condition is that the pair  $\{A, B\}$  is controllable, that is, the **controllability matrix**

$$\mathcal{Q}_C \triangleq [B \ AB \ \cdots \ A^{n_x-1}B] \quad (1.3.7-5)$$

has **full rank**, which in this case is  $n_x$ , the lower of its two dimensions (the other is  $n_x n_u$ ).

## Observability

A continuous-time (deterministic) system is **completely observable** if its initial state can be *fully and uniquely* recovered from its output, observed over a finite time interval, and the knowledge of the input.

Note that since the system is deterministic, knowledge of the initial state is equivalent to knowledge of the state at any time. Thus, using the output (1.3.7-2), which is in general a vector of dimension  $n_z < n_x$ , and the input, one can then recover the state perfectly.

For a *linear time-invariant system*, the observability condition is that the pair  $\{A, C\}$  is observable; that is, the **observability matrix**

$$\mathcal{Q}_O \triangleq \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n_x-1} \end{bmatrix} \quad (1.3.7-6)$$

has full rank  $n_x$ .

## Example

Consider the system (double integrator)

$$\ddot{\xi}(t) = u(t) \quad (1.3.7-7)$$

With the state vector consisting of the position  $\xi$  and velocity  $\dot{\xi}$

$$x = [\xi \ \dot{\xi}]' \quad (1.3.7-8)$$

the state space representation of this system is

$$\dot{x}(t) = Ax(t) + Bu(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \quad (1.3.7-9)$$

The controllability matrix is

$$\mathcal{Q}_C = [B \ AB] = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (1.3.7-10)$$

which has full rank  $n_x = 2$ , i.e., the state is *completely controllable*. While the input  $u$  enters directly only into the velocity component of the state, it does have the capability of controlling both state components. If the observation is

$$z(t) = Cx(t) = [1 \ 0]x(t) \quad (1.3.7-11)$$

i.e., the position  $\xi$ , the observability matrix is

$$\mathcal{Q}_O = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (1.3.7-12)$$

and it has full rank  $n_x = 2$ , i.e., the state is *completely observable*.

If the observation is

$$z(t) = Cx(t) = [0 \ 1]x(t) \quad (1.3.7-13)$$

i.e., the velocity  $\dot{\xi}$ , the observability matrix is

$$\mathcal{Q}_O = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (1.3.7-14)$$

and it has rank  $1 < n_x = 2$ , i.e., the state is *not completely observable*. The reason is that, while the position observations (1.3.7-11) provide information about the velocity, the velocity observations (1.3.7-13) do not provide any position information.

### 1.3.8 Discrete-Time Linear Dynamic Systems — Controllability and Observability

#### The State-Space Representation

Deterministic linear dynamic systems can be described in discrete time by a state equation of the form

$$x(k+1) = F(k)x(k) + G(k)u(k) \quad (1.3.8-1)$$

where

- $x(k)$  is the state of the system, a vector of dimension  $n_x$ ,
- $F(k)$  is the transition matrix ( $n_x \times n_x$ ),
- $u(k)$  is the input (control), a vector of dimension  $n_u$ ,
- $G(k)$  is the input gain ( $n_x \times n_u$  matrix),

all at time  $k$ .

The output equation is

$$z(k) = H(k)x(k) \quad (1.3.8-2)$$

where

$z(k)$  is the output vector (observation or measurement), of dimension  $n_z$ ,

$H(k)$  is the measurement matrix ( $n_z \times n_x$ ).

The **state** of the system is the smallest dimension vector that completely summarizes the past of the system. Then the output at time  $j > k$  can be determined fully from  $x(k)$  and the intervening inputs

$$z(j) = z[j, x(k), u(k), \dots, u(j-1)] \quad (1.3.8-3)$$

A discrete-time system is **linear time-invariant** if it is described by (1.3.8-1) and (1.3.8-2) with  $F(k) = F$ ,  $G(k) = G$ , and  $H(k) = H$ .

## Controllability

A discrete-time (deterministic) system is **completely controllable** if, given an *arbitrary destination point* in the state space, there is an input sequence that will bring the system from any initial state to this point in a finite number of steps.

For a *linear time-invariant system*, the controllability condition is that the pair  $\{F, G\}$  is controllable; that is, the **controllability matrix**

$$Q_C \triangleq [G \ FG \ \cdots \ F^{n_x-1}G] \quad (1.3.8-4)$$

has full rank  $n_x$ .

## Observability

A (deterministic) system is **completely observable** if its initial state can be *fully and uniquely* recovered from a finite number of observations of its output and the knowledge of its input.

Note that since the system is deterministic, knowledge of the initial state is equivalent to knowledge of the state at any time. Thus, using the input and the output (1.3.8-2), which is in general a vector of dimension  $n_z < n_x$ , one can, *in the absence of noise*, recover the state perfectly.

For a *linear time-invariant system*, the observability condition is that the pair  $\{F, H\}$  is observable; that is, the **observability matrix**

$$Q_O \triangleq \begin{bmatrix} H \\ HF \\ \vdots \\ HF^{n_x-1} \end{bmatrix} \quad (1.3.8-5)$$

has full rank  $n_x$ .

## 1.4 BRIEF REVIEW OF PROBABILITY THEORY

### 1.4.1 Events and the Axioms of Probability

Consider an “experiment,” or, in general, a process with random outcomes. An **event** is a collection (set) of such outcomes — it is said to have occurred if the outcome is one of the elements of this set.

Denote by  $A$  an event in such an experiment (e.g., “even” or “5” in a die rolling experiment). Let  $S$  be the **sure event** in the experiment (e.g., “any number between 1 and 6” in the die rolling). Then the **probability** of an event is a number (**measure**) that satisfies the following three **axioms of probability**:

1. It is nonnegative

$$P\{A\} \geq 0 \quad \forall A \quad (1.4.1-1)$$

2. It is unity for the sure event

$$P\{S\} = 1 \quad (1.4.1-2)$$

3. It is additive over the union of **mutually exclusive** events; that is, if the events  $A$  and  $B$  have no common elements (their set intersection is  $\emptyset$ , the empty set)

$$A \cap B \triangleq \{A \text{ and } B\} \triangleq \{A, B\} = \emptyset \quad (1.4.1-3)$$

then their union (logical “or”) has probability

$$P\{A \cup B\} \triangleq P\{A \text{ or } B\} \triangleq P\{A + B\} = P\{A\} + P\{B\} \quad (1.4.1-4)$$

From the above it follows that

$$P\{A\} = 1 - P\{\bar{A}\} \leq 1 \quad (1.4.1-5)$$

where the overbar denotes the **complementary event**.

Since  $\emptyset$  is the complement of  $S$ , one has

$$P\{\emptyset\} = 0 \quad (1.4.1-6)$$

The event  $\emptyset$  is called the **impossible event**.

### Extension

The extended version of Axiom 3 such that the probability is additive over the union of an *infinite number of mutually exclusive events* — **countable additivity** — is necessary when an experiment has an infinite number of outcomes. This is a key point in the definition of continuous-valued random variables.

## Remarks

**Relative-frequency** or **measure-of-belief** interpretation of probability are alternative, even though not rigorous, ways of introducing the concept of probability.

In practice one can use the relative frequency or measure of belief to assign, based on intuition (or engineering common sense), probabilities to certain events. This can be of major importance in certain engineering systems.

Note that probability is a scalar quantity that is dimensionless (in a physical sense). This is in contrast to, for instance, a probability density function (pdf) which, while a scalar, has in general a *physical dimension*.

## Example

For a die-rolling experiment, the space of outcomes is

$$S = \{f_1, \dots, f_6\} \quad (1.4.1-7)$$

where  $f_i$  denotes “face  $i$ ”,  $i = 1, \dots, 6$ .

Events are subsets of  $S$ , for example,

$$F_1 = \{f_1\} \quad (1.4.1-8)$$

$$\{i = \text{even}\} = \{f_2, f_4, f_6\} \quad (1.4.1-9)$$

$$\{2 < i \leq 4\} = \{f_3, f_4\} \quad (1.4.1-10)$$

The sure event

$$\{\text{any face}\} = S \quad (1.4.1-11)$$

is denoted by the same symbol as the space of outcomes since it is the set of all possible outcomes.

An impossible event is, for example,

$$\{i \leq 1 \text{ and even}\} = \emptyset \quad (1.4.1-12)$$

This is clearly an empty subset of  $S$ .

Probabilities are assigned (postulated) for each element of  $S$  (each outcome). For a finite number of outcomes this *granular approach* yields, using Axiom 3, the probabilities of all the subsets of  $S$ , i.e., the events.

Postulating a fair die — each outcome (face) is equally probable — one has from Axioms 2 and 3,

$$1 = P\{S\} = P\left\{\bigcup_{i=1}^6 F_i\right\} = \sum_{i=1}^6 P\{F_i\} = 6P\{F_i\} \quad (1.4.1-13)$$

which yields

$$P\{F_i\} = 1/6 \quad i = 1, \dots, 6 \quad (1.4.1-14)$$

From this it follows with Axiom 3 that

$$P\{i = \text{even}\} = P\{F_2\} + P\{F_4\} + P\{F_6\} = 1/2 \quad (1.4.1-15)$$

## 1.4.2 Random Variables and Probability Density Function

A scalar **random variable** is a (real-valued) function that assumes a certain *value* according to the outcome of a *random experiment*. The value taken by a random variable is called its **realization**.

The **probability density function (pdf)** of a scalar continuous-valued random variable  $x$  at  $x = \xi$  is

$$p_x(\xi) = \lim_{d\xi \rightarrow 0} \frac{P\{\xi - d\xi < x \leq \xi\}}{d\xi} \geq 0 \quad (1.4.2-1)$$

where  $P\{\cdot\}$  is the probability of the event  $\{\cdot\}$ .

The more common notation

$$p_x(x) = p(x) \quad (1.4.2-2)$$

where the argument defines the function, is used instead of (1.4.2-1). Also the term **density** is sometimes used instead of *pdf*.

From (1.4.2-1) and Axiom 3 from Subsection 1.4.1 it follows that

$$P\{\eta < x \leq \xi\} = \int_{\eta}^{\xi} p(x) dx \quad (1.4.2-3)$$

The function

$$P_x(\xi) = P\{x \leq \xi\} = \int_{-\infty}^{\xi} p(x) dx \quad (1.4.2-4)$$

is called the **cumulative probability distribution function (cdf)** of  $x$  at  $\xi$ . This is usually referred to as **distribution**.

Since the event  $\{x \leq \infty\}$  is the “sure” event, one has

$$P\{x \leq \infty\} = \int_{-\infty}^{\infty} p(x) dx = 1 \quad (1.4.2-5)$$

A pdf has to have the **normalization property** (1.4.2-5) that its total **probability mass** is unity — otherwise it is not a **proper density**.

The relationship between the density and the cumulative distribution is, from (1.4.2-4),

$$p(x) = \left. \frac{d}{d\xi} P_x(\xi) \right|_{\xi=x} \quad (1.4.2-6)$$

if the derivative exists.

An outline of the rigorous way of introducing the pdf is as follows:

1. First define the “basic” events  $\{x \leq \xi\}$ .
2. Show that from these events one can obtain via (a countable number of) set operations all the events of interest (e.g.,  $\{a < x \leq \xi\}$ ,  $\{x = a\}$ ).
3. Define the probabilities of the basic events, from which the probabilities of the other events of interest can be computed using the extended version of the third axiom of probability.
4. Finally, the pdf follows from (1.4.2-6) under suitable conditions of differentiability; note that (1.4.2-6) is equivalent to (1.4.2-1).

## Remarks

Note from (1.4.2-1) that with the numerator being dimensionless, the pdf of the random variable  $x$  has as its *physical dimension* the inverse of the physical dimension of  $x$ . (See also problem 1-5.)

The event  $\{x = a\}$  has, for a continuous-valued random variable, probability zero (even though it is *not impossible!*). This follows from (1.4.2-3) when the interval length tends to zero, in which case, as long as the density is finite, the integral is zero. Finite density means “no point masses” — this is discussed in the next section.

**Improper densities** — which do not integrate to unity (actually, their integral is not defined) — can be used, however, in certain circumstances.

## Example

A **uniformly distributed random variable** in the interval  $[a, b]$ , symbolically denoted as

$$x \sim \mathcal{U}(a, b) \quad (1.4.2-7)$$

has the pdf

$$p(x) = \mathcal{U}(x; a, b) \triangleq \begin{cases} \frac{1}{b-a} & x \in [a, b] \\ 0 & \text{elsewhere} \end{cases} \quad (1.4.2-8)$$

Its cdf is

$$P_x(\xi) = \begin{cases} 0 & \xi < a \\ \frac{\xi - a}{b - a} & \xi \in [a, b] \\ 1 & \xi > b \end{cases} \quad (1.4.2-9)$$

A **Gaussian (normal) random variable** with parameters  $\mu, \sigma^2$  (the mean and variance, respectively, discussed later) is symbolically denoted as

$$x \sim \mathcal{N}(\mu, \sigma^2) \quad (1.4.2-10)$$

Its pdf is

$$p(x) = \mathcal{N}(x; \mu, \sigma^2) \triangleq \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (1.4.2-11)$$

and the cdf is

$$\begin{aligned} P_x(\xi) &= \int_{-\infty}^{\xi} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = \int_{-\infty}^{\frac{\xi-\mu}{\sigma}} \frac{1}{\sqrt{2\pi}} e^{-\frac{\eta^2}{2}} d\eta \\ &= \int_{-\infty}^{\frac{\xi-\mu}{\sigma}} \mathcal{N}(\eta; 0, 1) d\eta \triangleq \mathcal{G}\left(\frac{\xi - \mu}{\sigma}\right) \end{aligned} \quad (1.4.2-12)$$

where  $\mathcal{G}$  is the **cumulative standard Gaussian distribution** (with mean  $\mu = 0$  and standard deviation  $\sigma = 1$ ). Its argument above is the *number of standard deviations away from the mean*. This integral has been extensively tabulated (see, e.g., Table 1.5.4-1).

### Note

The symbols  $\mathcal{U}$  and  $\mathcal{N}$  with two arguments indicate “*distributed as*,” while with three arguments they denote the corresponding pdf.

### 1.4.3 Probability Mass Function

The **probability mass function (pmf)** of a scalar random variable  $x$ , which can take values in the set  $\{\xi_i, i = 1, \dots, n\}$  (i.e., it is discrete-valued), is

$$\mu_x(\xi_i) = P\{x = \xi_i\} = \mu_i \quad i = 1, \dots, n \quad (1.4.3-1)$$

where  $\mu_i$  are the **point masses**.

Similarly to (1.4.2-5), the requirement for a **proper pmf** is

$$\sum_{i=1}^n \mu_i = 1 \quad (1.4.3-2)$$

Using the **Dirac (impulse) delta function** defined by

$$\delta(x) = 0 \quad \forall x \neq 0 \quad (1.4.3-3)$$

and

$$\int_{-\infty}^{\infty} \delta(x) dx = 1 \quad (1.4.3-4)$$

one can write a pdf corresponding to (1.4.3-1) as

$$p(x) = \sum_{i=1}^n \mu_i \delta(x - \xi_i) \quad (1.4.3-5)$$

Note that the above satisfies the **normalization property** (1.4.2-5) of a pdf in that it integrates to unity.

The distribution corresponding to the above density, called **cumulative probability mass function (cpmf)**, has jumps at  $\xi_i$  — it is a staircase function: Its derivative is zero everywhere except at the jumps where it is an impulse function.

The expression of the cpmf can be written, in terms of the **unit step function**  $1(\cdot)$ , as

$$P\{x \leq \xi\} = \sum_{i=1}^n \mu_i 1(\xi - \xi_i) \quad (1.4.3-6)$$

**Note**

For a random variable with no point mass at  $\xi$ , one has

$$P\{x \leq \xi\} = P\{x < \xi\} \quad (1.4.3-7)$$

**Example**

The **Poisson pmf** with rate  $\lambda$

$$P\{x = n\} = e^{-\lambda T} \frac{(\lambda T)^n}{n!} \quad n = 0, 1, \dots \quad (1.4.3-8)$$

describes the number of random points in an interval  $T$  can also be written as

$$p(x) = \sum_{n=0}^{\infty} e^{-\lambda T} \frac{(\lambda T)^n}{n!} \delta(x - n) \quad (1.4.3-9)$$

**1.4.4 Mixed Random Variable and Mixed Probability-PDF**

A **mixed random variable** or a **hybrid random variable**  $x$  is one which can take values in a continuous set  $X$  as well as over a discrete set of points  $\{\xi_i, i = 1, \dots, n\}$ . Such a random variable has a pdf of the form

$$p(x) = p_c(x) + \sum_{i=1}^n \mu_i \delta(x - \xi_i) \quad (1.4.4-1)$$

where  $p_c(x)$  is the continuous part of the pdf and  $\mu_i$  are the point masses.

Then

$$\int_{-\infty}^{\infty} p(x) dx = \int_{x \in X} p_c(x) dx + \sum_{i=1}^n \mu_i = 1 \quad (1.4.4-2)$$

The **joint probability-pdf of an event and a random variable** is defined as

$$P_{A,x}[A, \xi] = \lim_{d\xi \rightarrow 0} \frac{P\{A, \xi - d\xi < x \leq \xi\}}{d\xi} \quad (1.4.4-3)$$

and is denoted without subscripts as  $P[A, x]$  where the arguments define the function.

The following notations will be observed<sup>6</sup> in the sequel:

$$P\{\cdot\} = \text{probability of an event} \quad (1.4.4-4)$$

$$p(\cdot) \text{ or } p[\cdot] = \text{pdf} \quad (1.4.4-5)$$

$$\mu(\cdot) = \text{pmf} \quad (1.4.4-6)$$

$$P[\cdot, \cdot] = \text{mixed (joint) probability-pdf} \quad (1.4.4-7)$$

Some of the uncertainties in estimation problems are naturally modeled as mixed random variables.

<sup>6</sup>Most of the time. Some of the time we will be guilty of using “dynamic notation” — the same variable keeps being denoted by new symbols, and the same symbol is used for different variables.

### 1.4.5 Expectations and Moments of a Scalar Random Variable

The **expected value** of a scalar random variable, also called its **mean**, **average**, or **first moment**, is

$$E[x] = \int_{-\infty}^{\infty} xp(x) dx \triangleq \bar{x} \quad (1.4.5-1)$$

The  $n$ th **moment** is

$$E[x^n] = \int_{-\infty}^{\infty} x^n p(x) dx \quad (1.4.5-2)$$

The **second central moment** or **variance** is

$$\text{var}(x) \triangleq E[(x - \bar{x})^2] = \int_{-\infty}^{\infty} (x - \bar{x})^2 p(x) dx = E[x^2] - (\bar{x})^2 \triangleq \sigma_x^2 \quad (1.4.5-3)$$

The square root  $\sigma$  of the variance is called the **standard deviation**.

A random variable  $x$  with mean  $\bar{x}$  and variance  $\sigma^2$  will be denoted as

$$x \sim [\bar{x}, \sigma^2] \quad (1.4.5-4)$$

Note that this is similar to (1.4.2-10) except that the distribution is not specified.

The second moment, which is the **mean square (MS)** value, is equal to the square of the mean plus the variance

$$E[x^2] = [E(x)]^2 + \text{var}(x) = \bar{x}^2 + \sigma_x^2 \quad (1.4.5-5)$$

For a zero-mean random variable, the standard deviation is the **root mean square (RMS)** value.<sup>7</sup>

The expected value of a function  $g(x)$  of the random variable  $x$  is

$$E[g(x)] = \int_{-\infty}^{\infty} g(x)p(x) dx \quad (1.4.5-6)$$

that is, the (Lebesgue) integral with respect to the measure  $p(x) dx = dP(x)$ .

#### Examples

If  $x$  is uniformly distributed in the interval  $[a, b]$ , that is,

$$p(x) = \mathcal{U}(x; a, b) \triangleq \begin{cases} \frac{1}{b-a} & x \in [a, b] \\ 0 & \text{elsewhere} \end{cases} \quad (1.4.5-7)$$

then one has

$$E[x] = \frac{b+a}{2} \quad \text{var}(x) = \frac{(b-a)^2}{12} \quad (1.4.5-8)$$

---

<sup>7</sup>Not to be confused with the RMS value of a waveform.

(See also problem 1-6.)

For a Gaussian random variable with parameters  $\mu$  and  $\sigma^2$  (see 1.4.2-10) the mean is

$$\begin{aligned} E[x] &= \int_{-\infty}^{\infty} x \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx \\ &= \int_{-\infty}^{\infty} (x - \mu) \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx + \mu \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx \\ &= \mu \end{aligned} \quad (1.4.5-9)$$

where in the last line above the first integral is zero due to symmetry and the second is unity due to the pdf normalization property.

The variance is

$$\begin{aligned} \text{var}(x) &= E[(x - \mu)^2] = \int_{-\infty}^{\infty} (x - \mu)^2 \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx \\ &= \int_{-\infty}^{\infty} x^2 \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} dx \end{aligned} \quad (1.4.5-10)$$

The last integral above can be evaluated by differentiating w.r.t.  $\sigma$  the identity

$$\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} dx = \sigma \quad (1.4.5-11)$$

which follows from the pdf normalization property. Thus

$$\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma^3} x^2 e^{-\frac{x^2}{2\sigma^2}} dx = 1 \quad (1.4.5-12)$$

Using this in (1.4.5-10) yields

$$\text{var}(x) = \sigma^2 \quad (1.4.5-13)$$

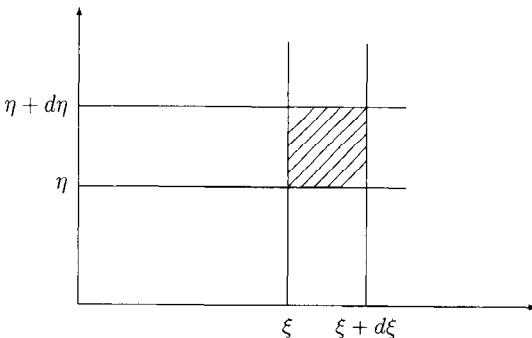
The mean of the Poisson random variable  $x$  from (1.4.3-9) is

$$E[x] = \int_{-\infty}^{\infty} x \sum_{n=0}^{\infty} e^{-\lambda T} \frac{(\lambda T)^n}{n!} \delta(x-n) dx = \sum_{n=0}^{\infty} n e^{-\lambda T} \frac{(\lambda T)^n}{n!} = \lambda T \quad (1.4.5-14)$$

## 1.4.6 Joint PDF of Two Random Variables

The **joint pdf** of two random variables  $x$  and  $y$  is defined in terms of the probability of the following joint event, denoted with the set intersection symbol (see Fig. 1.4.6-1), namely,

$$p_{x,y}(\xi, \eta) \triangleq \lim_{d\xi \rightarrow 0, d\eta \rightarrow 0} \frac{P\{\{\xi - d\xi < x \leq \xi\} \cap \{\eta - d\eta < y \leq \eta\}\}}{d\xi d\eta} \quad (1.4.6-1)$$



**Figure 1.4.6-1:** Event for definition of a joint pdf.

Integrating the joint pdf of two random variables over one of the variables yields the pdf of the other random variable

$$\int_{-\infty}^{\infty} p_{x,y}(\xi, \eta) d\eta = p_x(\xi) \quad (1.4.6-2)$$

or, using the simpler notation (1.4.2-2),

$$\int_{-\infty}^{\infty} p(x, y) dy = p(x) \quad (1.4.6-3)$$

The resulting pdf, which pertains to a single random variable (introduced in Subsection 1.4.2), is also called **marginal pdf** or **marginal density**.

Similarly to (1.4.2-4), the **joint cdf** is

$$P_{x,y}(\xi, \eta) = P\{x \leq \xi, y \leq \eta\} = \int_{x=-\infty}^{\xi} \int_{y=-\infty}^{\eta} p_{x,y}(x, y) dx dy \quad (1.4.6-4)$$

### Covariance and Correlation Coefficient

The **covariance of two scalar random variables**  $x_1$  and  $x_2$  with means  $\bar{x}_1$  and  $\bar{x}_2$ , respectively, is

$$\begin{aligned} \text{cov}(x_1, x_2) &\triangleq E[(x_1 - \bar{x}_1)(x_2 - \bar{x}_2)] \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x_1 - \bar{x}_1)(x_2 - \bar{x}_2) p(x_1, x_2) dx_1 dx_2 \triangleq \sigma_{x_1 x_2}^2 \end{aligned} \quad (1.4.6-5)$$

The **correlation coefficient** of these two random variables is the normalized quantity

$$\rho_{12} \triangleq \frac{\sigma_{x_1 x_2}^2}{\sigma_{x_1} \sigma_{x_2}} \quad (1.4.6-6)$$

where  $\sigma_{x_i}$  is the standard deviation of  $x_i$ .

Due to the normalization, the *magnitude of the correlation coefficient* of any two random variables obeys the following inequality:

$$|\rho_{12}| \leq 1 \quad (1.4.6-7)$$

It can be shown that this is a consequence of the Schwarz inequality (1.3.4-2) for random variables (where the inner product is the covariance and the norm squared is the variance — this is discussed in Section 3.3).

Two random variables whose correlation coefficient is zero are said to be *uncorrelated*. Clearly, the uncorrelatedness of two random variables is equivalent to their covariance being zero. Furthermore, the random variables  $x_1$  and  $x_2$  are uncorrelated if and only if

$$E[x_1 x_2] = E[x_1] E[x_2] \quad (1.4.6-8)$$

At the other extreme, if the correlation coefficient of two random variables has magnitude unity, then it can be shown that they are linearly dependent (i.e., one is a linear function of the other). Two random variables  $x_1$  and  $x_2$  are said to be *linearly dependent* if

$$a_1 x_1 + a_2 x_2 = 0 \quad (1.4.6-9)$$

for some  $a = [a_1 \ a_2]' \neq 0$ .

### Example

Consider the random variables  $x$  and  $y$  with means  $\bar{x}$ ,  $\bar{y}$  and variances  $\sigma_x^2$ ,  $\sigma_y^2$ , respectively, and uncorrelated from each other (i.e.,  $\text{cov}(x, y) = 0$ ).

Let

$$z = ax + by \quad (1.4.6-10)$$

The mean of  $z$  is

$$\bar{z} = E[z] = E[ax + by] = aE[x] + bE[y] = a\bar{x} + b\bar{y} \quad (1.4.6-11)$$

while its variance is

$$\begin{aligned} \text{var}(z) &= E[(z - \bar{z})^2] = E[(ax + by - a\bar{x} - b\bar{y})^2] \\ &= E\{[a(x - \bar{x}) + b(y - \bar{y})]^2\} \\ &= a^2 E[(x - \bar{x})^2] + b^2 E[(y - \bar{y})^2] + 2ab E[(x - \bar{x})(y - \bar{y})] \\ &= a^2 \sigma_x^2 + b^2 \sigma_y^2 \end{aligned} \quad (1.4.6-12)$$

since the cross-term — the covariance of  $x$  and  $y$  — is zero.

The covariance of  $z$  and  $x$  is

$$\begin{aligned} E[(z - \bar{z})(x - \bar{x})] &= E\{[a(x - \bar{x}) + b(y - \bar{y})](x - \bar{x})\} \\ &= a\sigma_x^2 + b \text{cov}(x, y) = a\sigma_x^2 \end{aligned} \quad (1.4.6-13)$$

### 1.4.7 Independent Events and Independent Random Variables

Two events are *independent* if the probability of their joint event equals the product of their marginal probabilities

$$P\{A \cap B\} \triangleq P\{A, B\} = P\{A\}P\{B\} \quad (1.4.7-1)$$

A set of  $n$  events  $A_i, i = 1, \dots, n$  are independent if the (joint) probability of their intersection is equal to the product of the corresponding (marginal) event probabilities

$$P\left\{\bigcap_{i=1}^n A_i\right\} = \prod_{i=1}^n P\{A_i\} \quad (1.4.7-2)$$

and the same property holds also for any subset of these events.

Similarly,  $n$  random variables are *independent* if their joint pdf equals the product of their corresponding marginal densities

$$p(x_1, \dots, x_n) = \prod_{i=1}^n p(x_i) \quad (1.4.7-3)$$

A set of random variables is called *independent, identically distributed (i.i.d.)* if (1.4.7-3) holds and their marginal distributions (or densities) are identical.

The pdf of the sum of two independent random variables

$$y = x_1 + x_2 \quad (1.4.7-4)$$

is the *convolution* of their marginal densities

$$p(y) = \int p_{x_1}(y - x)p_{x_2}(x) dx \quad (1.4.7-5)$$

(See also problem 1-1.)

### 1.4.8 Vector-Valued Random Variables and Their Moments

The pdf of the *vector-valued random variable*

$$x = [x_1 \ \cdots \ x_n]' \quad (1.4.8-1)$$

at

$$\xi = [\xi_1 \ \cdots \ \xi_n]' \quad (1.4.8-2)$$

is defined as the *joint density* of its components

$$p_{x_1, \dots, x_n}(\xi_1, \dots, \xi_n) \triangleq p_x(\xi) \triangleq \lim_{d\xi_1 \rightarrow 0, \dots, d\xi_n \rightarrow 0} \frac{P\{\bigcap_{i=1}^n \{\xi_i - d\xi_i < x_i \leq \xi_i\}\}}{d\xi_1 \cdots d\xi_n} \quad (1.4.8-3)$$

where the set intersection symbol  $\cap$  is used to denote a joint event

$$A \cap B \triangleq \{A \text{ and } B\} \triangleq \{A, B\} \quad (1.4.8-4)$$

The shorter notation  $p(x)$  will be used if it does not lead to ambiguities.

The mean of the vector  $x$  is the result of the  $n$ -fold integration

$$E[x] = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \xi p_x(\xi) d\xi_1 \cdots d\xi_n = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} xp(x) dx_1 \cdots dx_n \triangleq \bar{x} \quad (1.4.8-5)$$

The **covariance matrix** of the  $n$ -vector  $x$  is obtained from the  $n$ -fold integration (written with the short notation and with the limits omitted)

$$\text{cov}(x) \triangleq E[(x - \bar{x})(x - \bar{x})'] = \int (x - \bar{x})(x - \bar{x})' p(x) dx \triangleq P_{xx} \quad (1.4.8-6)$$

where  $dx \triangleq \prod dx_i$ . Note that the covariance matrix is a *symmetric*  $n \times n$  matrix.

The diagonal elements of the *covariance matrix* are the *variances* of the components of  $x$ , while the off-diagonal elements are the (*scalar*) *covariances* between its components, as in (1.4.6-5).

The *covariance matrix* is *positive definite* (and thus nonsingular) unless there is a linear dependence among the components of  $x$ . If there is such a dependence, the covariance matrix is positive semidefinite.

## Note

Covariance matrices will be denoted by the symbol  $P$ , with a subscript or argument (in parentheses); the designation of a probability will be  $P\{\cdot\}$  where braces are used for the event it pertains to, as in (1.4.4-4).

## Characteristic Function and Moments

The **characteristic function** of a vector random variable is defined as the  $n$ -fold integral (with the limits, which are, as in (1.4.8-5), omitted)

$$M_x(s) = E[e^{s'x}] = \int e^{s'x} p(x) dx \quad (1.4.8-7)$$

which is the ( $n$ -dimensional) Fourier transform of the pdf with argument the (purely imaginary) vector  $s$  of the same dimension  $n$  as that of  $x$ .

The first moment of  $x$  can be obtained from the characteristic function as

$$E[x] = \nabla_s M_x(s)|_{s=0} \quad (1.4.8-8)$$

that is, its gradient is evaluated at  $s = 0$ , where  $\nabla_s$  is the (column) gradient operator

$$\nabla_s = \left[ \frac{\partial}{\partial s_1} \cdots \frac{\partial}{\partial s_n} \right]' \quad (1.4.8-9)$$

Similarly,

$$E[xx'] = \nabla_s \nabla'_s M_x(s)|_{s=0} \quad (1.4.8-10)$$

and so forth.

Due to this property, the characteristic function is also called the **moment-generating function**.

### Example

The characteristic function of the scalar Gaussian random variable

$$x \sim \mathcal{N}(\mu, \sigma^2) \quad (1.4.8-11)$$

is

$$M_x(s) = e^{s\mu + \frac{1}{2}s^2\sigma^2} \quad (1.4.8-12)$$

To find the moments of  $x$  it is convenient to deal with the zero-mean random variable

$$y = x - \mu \quad (1.4.8-13)$$

The moments of  $x$  follow easily from the moments of  $y$ , according to

$$E[x^n] = E[(y + \mu)^n] \quad (1.4.8-14)$$

The characteristic function of  $y$  is

$$M_y(s) = e^{\frac{1}{2}s^2\sigma^2} \quad (1.4.8-15)$$

and

$$\frac{d}{ds} M_y(s) = s\sigma^2 e^{\frac{1}{2}s^2\sigma^2} \quad (1.4.8-16)$$

Evaluating the above at  $s = 0$  yields the first moment (mean) of  $y$  as 0.

Similarly,

$$\frac{d^2}{ds^2} M_y(s) = [\sigma^2 + (s\sigma^2)^2] e^{\frac{1}{2}s^2\sigma^2} \quad (1.4.8-17)$$

which, at  $s = 0$ , yields

$$E[y^2] = E[(x - \mu)^2] = \text{var}(x) = \sigma^2 \quad (1.4.8-18)$$

The third derivative of the characteristic function of  $y$  is

$$\frac{d^3}{ds^3} M_y(s) = [2s\sigma^4 + (\sigma^2 + s^2\sigma^4)s\sigma^2] e^{\frac{1}{2}s^2\sigma^2} = [3s\sigma^4 + s^3\sigma^6] e^{\frac{1}{2}s^2\sigma^2} \quad (1.4.8-19)$$

which yields zero as the third moment of  $y$ .

The fourth moment of  $y$  is given by

$$\frac{d^4}{ds^4} M_y(s) = [3\sigma^4 + 3s^2\sigma^6 + (3s\sigma^4 + s^3\sigma^6)s\sigma^2] e^{\frac{1}{2}s^2\sigma^2} \quad (1.4.8-20)$$

at  $s = 0$  as

$$E[y^4] = E[(x - \mu)^4] = 3\sigma^4 \quad (1.4.8-21)$$

### 1.4.9 Conditional Probability and PDF

The **conditional probability** of an event  $A$  given  $B$  is defined as

$$P\{A|B\} = \frac{P\{A, B\}}{P\{B\}} \quad (1.4.9-1)$$

For independent events, the above becomes the unconditional probability.

Similarly, the **conditional pdf** of one random variable given another random variable is, using the simpler notation as in (1.4.2-2), given by

$$p(x|y) = \frac{p(x, y)}{p(y)} \quad (1.4.9-2)$$

For an event conditioned on a random variable, one has

$$P\{A|x\} = \frac{P[A, x]}{p(x)} \quad (1.4.9-3)$$

The conditional pdf of a random variable  $x$ , given an event  $A$ , is

$$p(x|A) = \frac{P[A, x]}{P\{A\}} \quad (1.4.9-4)$$

where the numerator above is a mixed probability-pdf, as defined in (1.4.4-3).

#### Example

If the conditioning event is

$$A = \{x \leq a\} \quad (1.4.9-5)$$

then one gets the **truncated pdf**

$$p(x|x \leq a) = \frac{P[x, x \leq a]}{P\{x \leq a\}} = \begin{cases} \frac{p(x)}{P\{x \leq a\}} & \text{if } x \leq a \\ 0 & \text{elsewhere} \end{cases} \quad (1.4.9-6)$$

which is the original one restricted to  $x \leq a$  and suitably *renormalized*.

If  $z = x + y$  and the conditional pdf of  $y$  given  $x$  is  $p_{y|x}(\cdot)$ , then

$$p(z|x) = p_{y|x}(z - x) \quad (1.4.9-7)$$

If, in addition,  $x$  and  $y$  are independent, then, with  $p_y(\cdot)$  denoting the pdf of  $y$ , one has

$$p(z|x) = p_y(z - x) \quad (1.4.9-8)$$

(See also problem 1-1.)

Assume  $x \sim \mathcal{N}(\mu_x, \sigma_x^2)$ ,  $y \sim \mathcal{N}(\mu_y, \sigma_y^2)$  and independent. Then, with  $z = x + y$ ,

$$\begin{aligned} p(z|x) &= p_y(z-x) = \mathcal{N}(z-x; \mu_y, \sigma_y^2) \\ &= \frac{1}{\sqrt{2\pi}\sigma_y} e^{-\frac{(z-x-\mu_y)^2}{2\sigma_y^2}} = \mathcal{N}(z; x+\mu_y, \sigma_y^2) \end{aligned} \quad (1.4.9-9)$$

Note that the above can be viewed, due to the special form of the Gaussian density, as the pdf of  $z - x$  with mean  $\mu_y$  or the pdf of  $z$  with mean  $x + \mu_y$ .

If  $x \sim \mathcal{N}(\mu, \sigma^2)$ , then

$$p(x| |x - \mu| \leq a) = \begin{cases} \frac{1}{P\{|x - \mu| \leq a\}} \mathcal{N}(x; \mu, \sigma^2) & |x| \leq a \\ 0 & \text{elsewhere} \end{cases} \quad (1.4.9-10)$$

is a **truncated Gaussian**.

The normalization constant above is the probability that  $x - \mu$  lies in the closed interval<sup>8</sup>  $[-a, a]$ , which is

$$\begin{aligned} P\{|x - \mu| \leq a\} &= P\{x - \mu \in [-a, a]\} = P\{x \in [\mu - a, \mu + a]\} \\ &= P\{x \in (-\infty, \mu + a]\} - P\{x \in (-\infty, \mu - a)\} \\ &= \mathcal{G}\left(\frac{(\mu + a) - \mu}{\sigma}\right) - \mathcal{G}\left(\frac{(\mu - a) - \mu}{\sigma}\right) \\ &= \mathcal{G}\left(\frac{a}{\sigma}\right) - \mathcal{G}\left(\frac{-a}{\sigma}\right) = 1 - 2\mathcal{G}\left(\frac{-a}{\sigma}\right) \end{aligned} \quad (1.4.9-11)$$

where  $\mathcal{G}$  is the cumulative standard Gaussian distribution and the last equality follows from the symmetry of the Gaussian density. The last term above is the tail probability to the left of  $(-a)/\sigma$  or to the right of  $(a)/\sigma$ .

### 1.4.10 The Total Probability Theorem

Let the events  $B_i$ ,  $i = 1, \dots, n$ , be **mutually exclusive**, that is,

$$P\{B_i, B_j\} = 0 \quad \forall i \neq j \quad (1.4.10-1)$$

and **exhaustive**, that is,

$$\sum_{i=1}^n P\{B_i\} = 1 \quad (1.4.10-2)$$

Such a set of events is a *partition of the space of outcomes*. Then, the **total probability theorem** states that for any event  $A$ , its probability can be decomposed in terms of conditional probabilities as follows:

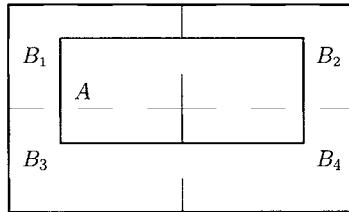
$$P\{A\} = \sum_{i=1}^n P\{A, B_i\} = \sum_{i=1}^n P\{A|B_i\}P\{B_i\} \quad (1.4.10-3)$$

<sup>8</sup>A bracket indicates that the interval is closed at the corresponding end, i.e., it includes the endpoint. A parenthesis indicates that the interval is open at the corresponding end, i.e., it excludes the endpoint.

The key to the above is the mutual exclusiveness and exhaustiveness of the set of events  $B_i$ ,  $i = 1, \dots, n$ .

Figure 1.4.10-1 illustrates this with the set theory counterpart, namely,

$$A = \bigcup_{i=1}^n (A \cap B_i) \quad (1.4.10-4)$$



**Figure 1.4.10-1:** The total probability theorem (set theory counterpart).

The version of the total probability theorem for random variables is

$$p(x) = \int_{-\infty}^{\infty} p(x, y) dy = \int_{-\infty}^{\infty} p(x|y)p(y) dy \quad (1.4.10-5)$$

Note that this is a combination of the definition of the conditional pdf (1.4.9-2) and property (1.4.6-3) that the joint pdf integrated out with respect to one random variable yields the marginal pdf of the other variable.

For mixed event/random variable situations, one has

$$p(x) = \sum_{i=1}^n p(x|B_i)P\{B_i\} \quad (1.4.10-6)$$

(this is illustrated in an example in Subsection 1.4.14) and

$$P\{A\} = \int_{-\infty}^{\infty} P\{A|x\}p(x) dx = E[P\{A|x\}] \quad (1.4.10-7)$$

where the expectation operator averages over  $x$ .

An additional *common* conditioning in *all* the probabilities is permissible, for instance,

$$P\{A|C\} = \sum_{i=1}^n P\{A, B_i|C\} = \sum_{i=1}^n P\{A|B_i, C\}P\{B_i|C\} \quad (1.4.10-8)$$

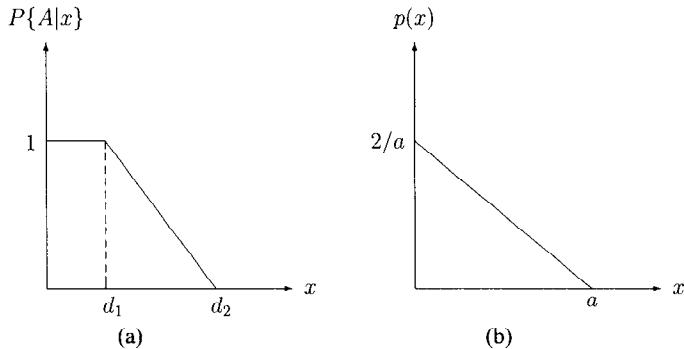
or

$$P\{A|y\} = \int_{-\infty}^{\infty} P\{A|x, y\}p(x|y) dx \quad (1.4.10-9)$$

Similarly,

$$p(x|z) = \int_{-\infty}^{\infty} p(x, y|z) dy = \int_{-\infty}^{\infty} p(x|y, z)p(y|z) dy \quad (1.4.10-10)$$

The total probability theorem is the primary tool in obtaining the state estimate in the presence of extraneous uncertainties (e.g., model uncertainties or measurements of uncertain origin).



**Figure 1.4.10-2:** Success probability and miss distance pdf.

### Example

The probability of event  $A$ , which denotes success in a hunting target shooting, can be modeled as a function of the “miss distance”  $x$  by the conditional probability (see Fig. 1.4.10-2a)

$$P\{A|x\} = \begin{cases} 1 & x \leq d_1 \\ \frac{x - d_2}{d_1 - d_2} & d_1 < x \leq d_2 \\ 0 & x > d_2 \end{cases} \quad (1.4.10-11)$$

The miss distance pdf can be taken as (see Fig. 1.4.10-2b)

$$p(x) = \begin{cases} \frac{2(a-x)}{a^2} & 0 < x \leq a \\ 0 & \text{otherwise} \end{cases} \quad (1.4.10-12)$$

Then, assuming  $d_2 < a$ , the (unconditional) probability of success is

$$\begin{aligned} P\{A\} &= \int_{-\infty}^{\infty} P\{A|x\}p(x) dx = \int_0^{d_1} p(x) dx + \int_{d_1}^{d_2} \frac{x - d_2}{d_1 - d_2} p(x) dx \\ &= P\{x \leq d_1\} + \int_{d_1}^{d_2} \frac{x - d_2}{d_1 - d_2} \frac{2(a-x)}{a^2} dx \\ &= \frac{d_1(2a - d_1)}{a^2} + \frac{(d_2 - d_1)}{3a^2} (3a - 2d_1 - d_2) \end{aligned} \quad (1.4.10-13)$$

### 1.4.11 Bayes' Formula

The probability of an event  $B_i$  conditioned on event  $A$  can be expressed in terms of the reverse conditioning as follows:

$$P\{B_i|A\} = \frac{P\{A|B_i\}P\{B_i\}}{P\{A\}} \quad (1.4.11-1)$$

This is known as **Bayes' formula** or **Bayes' theorem** and is also referred to as **Bayes' rule**.<sup>9</sup>

The conditional probability of  $B_i$  is sometimes referred to as **posterior probability**, while the unconditional one is referred to as **prior probability**.

Using the total probability theorem (1.4.10-3) for the denominator in the above, one also has (if  $B_j$ ,  $j = 1, \dots, n$  are mutually exclusive and exhaustive)

$$P\{B_i|A\} = \frac{P\{A|B_i\}P\{B_i\}}{\sum_{j=1}^n P\{A|B_j\}P\{B_j\}} = \frac{1}{c}P\{A|B_i\}P\{B_i\} \quad (1.4.11-2)$$

where now the denominator  $c$  appears clearly as the **normalizing constant**, which guarantees that

$$\sum_{i=1}^n P\{B_i|A\} = 1 \quad (1.4.11-3)$$

**Bayes' formula for random variables** is written as

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\int p(y|x)p(x) dx} \quad (1.4.11-4)$$

In this case the unconditional pdf  $p(x)$  is also called the **prior pdf** and the conditional pdf  $p(x|y)$  is also called the **posterior pdf**.

For a mixed case

$$P\{B_i|x\} = \frac{p(x|B_i)P\{B_i\}}{p(x)} = \frac{p(x|B_i)P\{B_i\}}{\sum_{j=1}^n p(x|B_j)P\{B_j\}} \quad (1.4.11-5)$$

where the denominator is again the normalizing factor.

When there are several conditioning random variables or events, Bayes' formula can be used to "switch" only some of them. For example,

$$P\{B_i|x, y\} = \frac{p(x|B_i, y)P\{B_i|y\}}{p(x|y)} = \frac{p(x|B_i, y)P\{B_i|y\}}{\sum_{j=1}^n p(x|B_j, y)P\{B_j|y\}} \quad (1.4.11-6)$$

These equations are the key tools in state estimation for **hybrid systems** — systems with continuous as well as discrete uncertainties.

## Remarks

The prior  $p(x)$  reflects the (possibly subjective<sup>10</sup>) initial **degree of belief**, which, when combined with the **evidence from the data** — the **likelihood function** of  $x$ , namely,  $p(y|x)$  — yields the posterior  $p(x|y)$ .

<sup>9</sup>**Bayes' decision rule** (or **Bayes' criterion** or **Bayes' principle**) in a decision problem is to minimize the expected value of a cost function, called **Bayes' risk**.

<sup>10</sup>Similarly to the well-known fact that the entire population of the world is divided into two groups (those who believe this and those who don't), the people dealing with probability are divided into two groups: proud Bayesians (who believe in priors) and closet Bayesians.

**Bayes' postulate** states that in the absence of prior knowledge, a **uniform prior pdf** should be chosen. While this is reasonable for discrete valued random variables, for continuous valued random variables this would imply a nonuniform pdf on any nonlinear function of this random variable and this has generated a lot of controversy among statisticians.

The use of a uniform prior pdf over an infinite interval, called **diffuse** or **noninformative**, also called **Dirichlet prior** is discussed in more detail in Subsection 2.3.4.

### Example

The “noise” in a device, denoted as  $x$  and modeled as a random variable, is observed. If the device is operating according to its specifications, then one has event  $B_1$ , defined as

$$B_1 = \{x \sim \mathcal{N}(0, \sigma_1^2)\} \quad (1.4.11-7)$$

The defective device event is assumed for simplicity defined as

$$B_2 = \{x \sim \mathcal{N}(0, \sigma_2^2)\} \quad (1.4.11-8)$$

The conditional probability that the device is defective given the observation  $x$  is

$$P\{B_2|x\} = \frac{p(x|B_2)P\{B_2\}}{p(x|B_1)P\{B_1\} + p(x|B_2)P\{B_2\}} \quad (1.4.11-9)$$

Using the explicit expressions of the Gaussian densities defining the events  $B_1$  and  $B_2$ , the above can be written, after cancellations, as

$$P\{B_2|x\} = \frac{1}{1 + \frac{\sigma_2}{\sigma_1} e^{-\frac{x^2}{2\sigma_1^2} + \frac{x^2}{2\sigma_2^2}} \frac{P\{B_1\}}{P\{B_2\}}} \quad (1.4.11-10)$$

Assume that a priori the two events are equiprobable

$$P\{B_1\} = \frac{1}{2} \quad P\{B_2\} = 1 - P\{B_1\} = \frac{1}{2} \quad (1.4.11-11)$$

If  $\sigma_2 = 4\sigma_1$  and the observation obtained is  $x = \sigma_2$ , then

$$P\{B_2|x\} = \frac{1}{1 + 4e^{-8+0.5}} \approx 0.998 \quad (1.4.11-12)$$

If the prior is different, namely,  $P\{B_2\} = 10^{-2}$ , then

$$P\{B_2|x\} = \frac{1}{1 + 396e^{-8+0.5}} \approx 0.82 \quad (1.4.11-13)$$

i.e., regardless of the prior, the “evidence from the data” indicates a very high probability that the device is defective.

### 1.4.12 Conditional Expectations and Their Smoothing Property

The **conditional expectation** is defined similarly to the unconditional expectation (1.4.5-1) but *with respect to a conditional pdf*, that is,

$$E[x|y] = \int_{-\infty}^{\infty} xp(x|y) dx \quad (1.4.12-1)$$

Similarly, for a function of the random variable  $x$  (and possibly  $y$ ), one has

$$E[g(x,y)|y] = \int_{-\infty}^{\infty} g(x,y)p(x|y) dx \quad (1.4.12-2)$$

Note that (1.4.12-1) and (1.4.12-2) are functions of the conditioning argument  $y$ .

The **smoothing property of the expectations** states that the expected value of a conditional expected value is the (unconditional) expected value:

$$\begin{aligned} E\{E[x|y]\} &= \int_{y=-\infty}^{\infty} \left[ \int_{x=-\infty}^{\infty} xp(x|y) dx \right] p(y) dy \\ &= \int_{x=-\infty}^{\infty} x \left[ \int_{y=-\infty}^{\infty} p(x,y) dy \right] dx \\ &= \int_{x=-\infty}^{\infty} xp(x) dx = E[x] \end{aligned} \quad (1.4.12-3)$$

In the first line of the above, the inside expectation is a function of  $y$ , which is “averaged out” (integrated over) by the outside expectation.

Equation (1.4.12-3) is also called the **law of iterated expectations**, summarized as

$$\boxed{E[E[x|y]] = E[x]} \quad (1.4.12-4)$$

The same property holds when the conditioning is on events or mixed random variables.

This is used in the evaluation of the performance of certain estimation algorithms operating in the presence of several types of uncertainties — it allows us to handle different types of uncertainties sequentially rather than simultaneously.

#### Example

To illustrate the smoothing property of expectations, consider the conditional mean

$$E(x|z) = \bar{x} + P_{xz}P_{zz}^{-1}(z - \bar{z}) \quad (1.4.12-5)$$

where  $x$  and  $z$  are jointly Gaussian (this is derived later in (1.4.14-17)); overbar denotes the unconditional expected values.

Applying the unconditional expectation operator on the above, where the only random variable on the right-hand side is  $z$ , with mean  $\bar{z}$ , yields

$$E[E(x|z)] = E[\bar{x}] + P_{xz}P_{zz}^{-1} E(z - \bar{z}) = \bar{x} = E[x] \quad (1.4.12-6)$$

### 1.4.13 Gaussian Random Variables

The pdf of a (scalar) **Gaussian** or **normal random variable** is

$$p(x) = \mathcal{N}(x; \bar{x}, \sigma^2) \triangleq \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\bar{x})^2}{2\sigma^2}} \quad (1.4.13-1)$$

where  $\mathcal{N}(\cdot)$  denotes the **normal pdf (density)** with argument  $x$ , mean  $\bar{x}$ , and variance  $\sigma^2$ . The first two moments, which fully characterize a Gaussian random variable, are referred to as its **statistics**.

Another notation equivalent to the above is

$$x \sim \mathcal{N}(\bar{x}, \sigma^2) \quad (1.4.13-2)$$

which states that  $x$  is *normally distributed* with the corresponding mean and variance. Note the different meanings of  $\mathcal{N}$  in (1.4.13-1) and (1.4.13-2) — they are, however, specified by their arguments: three in (1.4.13-1) vs. two in (1.4.13-2).

A **vector-valued Gaussian** random variable has the density

$$\mathcal{N}(x; \bar{x}, P) \triangleq |2\pi P|^{-1/2} e^{-\frac{1}{2}(x-\bar{x})'P^{-1}(x-\bar{x})} \quad (1.4.13-3)$$

where

$$\bar{x} = E[x] \quad (1.4.13-4)$$

$$P = E[(x - \bar{x})(x - \bar{x})'] \quad (1.4.13-5)$$

are, respectively, the mean and covariance matrix of the vector  $x$ .

The determinant in (1.4.13-3) has been written with the factor  $2\pi$  inside it by making use of (1.3.3-5). This avoids the need to indicate the dimension of the vector  $x$  in (1.4.13-3).

The components of a Gaussian distributed vector are said to be **jointly Gaussian**.

If the covariance matrix  $P$  is diagonal — that is, the components of the **Gaussian** random vector  $x$  are *uncorrelated* — then they are also *independent* because their joint pdf equals the product of the marginals.

A very important property of Gaussian random variables is that they *stay Gaussian under linear transformations*. Consequently, the sum of (jointly) Gaussian random variables is also Gaussian. This property is a major reason that the Gaussian model is very commonly used in estimation.

#### Example

If  $x_1$  and  $x_2$  are random vectors of the same dimension and jointly Gaussian distributed, that is,

$$p\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \mathcal{N}\left\{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}; \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix}, \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}\right\} \quad (1.4.13-6)$$

then

$$x = x_1 + x_2 \quad (1.4.13-7)$$

is Gaussian

$$p(x) = \mathcal{N}(x; \bar{x}, P) \quad (1.4.13-8)$$

where

$$\bar{x} = \bar{x}_1 + \bar{x}_2 \quad (1.4.13-9)$$

$$P = P_{11} + P_{12} + P_{21} + P_{22} \quad (1.4.13-10)$$

Symbolically, one can write, with notation (1.4.13-2),

$$\mathcal{N}(\bar{x}_1, P_{11}) + \mathcal{N}(\bar{x}_2, P_{22}) = \mathcal{N}(\bar{x}_1 + \bar{x}_2, P_{11} + P_{12} + P_{21} + P_{22}) \quad (1.4.13-11)$$

If  $x_1$  and  $x_2$  independent, the pdf of their sum  $x = x_1 + x_2$  conditioned on  $x_1$  is

$$\begin{aligned} p(x|x_1) &= p_{x_2}(x - x_1) \\ &= \mathcal{N}(x - x_1; \bar{x}_2, P_{22}) = |2\pi P_{22}|^{-1/2} e^{-\frac{1}{2}(x-x_1-\bar{x}_2)'P_{22}^{-1}(x-x_1-\bar{x}_2)} \\ &= \mathcal{N}(x; x_1 + \bar{x}_2, P_{22}) \end{aligned} \quad (1.4.13-12)$$

Note that in (1.4.13-12)  $x_1$  acts as a constant — it shifts the mean — since it is given in the conditioning.

#### 1.4.14 Joint and Conditional Gaussian Random Variables

Two random vectors  $x$  and  $z$  are **jointly Gaussian** if the **stacked vector**

$$y = \begin{bmatrix} x \\ z \end{bmatrix} \quad (1.4.14-1)$$

is Gaussian, that is,

$$p(x, z) = p(y) = \mathcal{N}(y; \bar{y}, P_{yy}) \quad (1.4.14-2)$$

The mean and covariance matrix of  $y$  in terms of those of  $x$  and  $z$  are

$$\bar{y} = \begin{bmatrix} \bar{x} \\ \bar{z} \end{bmatrix} \quad (1.4.14-3)$$

$$P_{yy} = \begin{bmatrix} P_{xx} & P_{xz} \\ P_{zx} & P_{zz} \end{bmatrix} \quad (1.4.14-4)$$

where

$$P_{xx} = \text{cov}(x) = E[(x - \bar{x})(x - \bar{x})'] \quad (1.4.14-5)$$

$$P_{zz} = \text{cov}(z) = E[(z - \bar{z})(z - \bar{z})'] \quad (1.4.14-6)$$

$$P_{xz} = \text{cov}(x, z) = E[(x - \bar{x})(z - \bar{z})'] = P'_{zx} \quad (1.4.14-7)$$

are the blocks of the **partitioned covariance matrix** (1.4.14-4).

The conditional pdf of  $x$  given  $z$  when they are jointly Gaussian is

$$p(x|z) = \frac{p(x,z)}{p(z)} = \frac{|2\pi P_{yy}|^{-1/2} e^{-\frac{1}{2}(y-\bar{y})'P_{yy}^{-1}(y-\bar{y})}}{|2\pi P_{zz}|^{-1/2} e^{-\frac{1}{2}(z-\bar{z})'P_{zz}^{-1}(z-\bar{z})}} \quad (1.4.14-8)$$

Advantage will be taken of the fact that the above is an exponential whose exponent is the difference of the exponents of the numerator and denominator.

The following notations will be used

$$\xi \triangleq x - \bar{x} \quad (1.4.14-9)$$

$$\zeta \triangleq z - \bar{z} \quad (1.4.14-10)$$

Using the new variables  $\xi$  and  $\zeta$ , the problem for the *nonzero mean random variables*  $x$  and  $y$  is reduced to that of *zero-mean variables*.

The exponent on the right-hand side of the conditional density (1.4.14-8) is (after multiplication by  $-2$ ) the quadratic form

$$\begin{aligned} q &= \begin{bmatrix} \xi \\ \zeta \end{bmatrix}' \begin{bmatrix} P_{xx} & P_{xz} \\ P_{zx} & P_{zz} \end{bmatrix}^{-1} \begin{bmatrix} \xi \\ \zeta \end{bmatrix} - \zeta' P_{zz}^{-1} \zeta \\ &= \begin{bmatrix} \xi \\ \zeta \end{bmatrix}' \begin{bmatrix} T_{xx} & T_{xz} \\ T_{zx} & T_{zz} \end{bmatrix} \begin{bmatrix} \xi \\ \zeta \end{bmatrix} - \zeta' P_{zz}^{-1} \zeta \end{aligned} \quad (1.4.14-11)$$

The relationships between the partitions of the inverse of the covariance matrix and the partitions of the original matrix are, using (1.3.3-7) to (1.3.3-10), given by

$$T_{xx}^{-1} = P_{xx} - P_{xz}P_{zz}^{-1}P_{zx} \quad (1.4.14-12)$$

$$P_{zz}^{-1} = T_{zz} - T_{zx}T_{xx}^{-1}T_{xz} \quad (1.4.14-13)$$

$$T_{xx}^{-1}T_{xz} = -P_{xz}P_{zz}^{-1} \quad (1.4.14-14)$$

The exponent (1.4.14-11) can be rewritten as

$$\begin{aligned} q &= \xi' T_{xx} \xi + \xi' T_{xz} \zeta + \zeta' T_{zx} \xi + \zeta' T_{zz} \zeta - \zeta' P_{zz}^{-1} \zeta \\ &= (\xi + T_{xx}^{-1} T_{xz} \zeta)' T_{xx} (\xi + T_{xx}^{-1} T_{xz} \zeta) + \zeta' (T_{zz} - T_{zx} T_{xx}^{-1} T_{xz}) \zeta - \zeta' P_{zz}^{-1} \zeta \\ &= (\xi + T_{xx}^{-1} T_{xz} \zeta)' T_{xx} (\xi + T_{xx}^{-1} T_{xz} \zeta) \end{aligned} \quad (1.4.14-15)$$

where use has been made of (1.4.14-13). The above procedure is called **completion of the squares** (actually, of the quadratic forms).

The result is a quadratic form in  $x$ , meaning that the conditional pdf of  $x$  given  $z$  is also Gaussian. This can be seen as follows: In view of (1.4.14-9), (1.4.14-10), and (1.4.14-14), the expression on the right-hand side of (1.4.14-15) is a quadratic form in

$$\xi + T_{xx}^{-1} T_{xz} \zeta = x - \bar{x} - P_{xz} P_{zz}^{-1} (z - \bar{z}) \quad (1.4.14-16)$$

From this one can recognize the conditional mean of  $x$  given  $z$  as

$$\boxed{E(x|z) \triangleq \hat{x} = \bar{x} + P_{xz}P_{zz}^{-1}(z - \bar{z})} \quad (1.4.14-17)$$

The corresponding conditional covariance is

$$\boxed{\text{cov}(x|z) \triangleq P_{xx|z} = T_{xx}^{-1} = P_{xx} - P_{xz}P_{zz}^{-1}P_{zx}} \quad (1.4.14-18)$$

Note that the conditional mean (1.4.14-17) is linear in the observation  $z$  and that the covariance (1.4.14-18) is independent of the observation.

The above are the *fundamental equations of linear estimation*.

### 1.4.15 Expected Value of Quadratic and Quartic Forms

Consider a vector-valued random variable  $x$  with mean  $\bar{x} = 0$  and covariance matrix  $P$ . Then, the *expected value of a quadratic form* with this random vector can be written as

$$E[x'Ax] = E[\text{tr}(x'Ax)] = E[\text{tr}(Axx')] = \text{tr}[AE(xx')] = \text{tr}(AP) \quad (1.4.15-1)$$

The same result can be obtained for a Gaussian random vector using its characteristic function as follows.

If

$$x \sim \mathcal{N}(\bar{x}, P) \quad (1.4.15-2)$$

then its characteristic (or moment generating) function is

$$M_x(s) = E[e^{s'x}] = e^{\frac{1}{2}s'Ps+s'\bar{x}} \quad (1.4.15-3)$$

Using the properties of the gradient, one can write

$$\begin{aligned} E[x'Ax] &= E[(\nabla_s e^{s'x})'Ax]|_{s=0} = E[\nabla'_s Ax e^{s'x}]|_{s=0} \\ &= \nabla'_s A E[x e^{s'x}]|_{s=0} = \nabla'_s A E[\nabla_s e^{s'x}]|_{s=0} \\ &= \nabla'_s A \nabla_s E[e^{s'x}]|_{s=0} = \nabla'_s A \nabla_s M_x(s)|_{s=0} \end{aligned} \quad (1.4.15-4)$$

Using the characteristic function (1.4.15-3) with  $\bar{x} = 0$  in the above yields (1.4.15-1) after the evaluation of the gradient. This technique, even though not the simplest for a quadratic form, can be used conveniently to evaluate the expected value of a quartic form, which is needed for the covariance of quadratic forms.

Note that the expected value of a quadratic form (1.4.15-1) holds regardless of the distribution of  $x$ .

For the *quartic form* to be considered next, the result will rely on the fact that  $x$  is Gaussian.

Analogously to (1.4.15-4), one has the **expected value of a quartic form** written as follows:

$$E[x'Ax x'Bx] = \nabla'_s A \nabla_s \nabla'_s B \nabla_s M_x(s)|_{s=0} \quad (1.4.15-5)$$

Using (1.4.15-3), it can be shown that after some computations the final result is

$$E[x'Ax x'Bx] = \text{tr}(AP)\text{tr}(BP) + 2\text{tr}(APBP) \quad (1.4.15-6)$$

The scalar version of the above for  $A = B = 1$  and  $P = \sigma^2$  is the well-known expression

$$E[x^4] = 3\sigma^4 \quad (1.4.15-7)$$

The **covariance of two quadratic forms** is, using (1.4.15-6),

$$\begin{aligned} E[[x'Ax - E(x'Ax)][x'Bx - E(x'Bx)]] &= E[x'Ax x'Bx] - E[x'Ax]E[x'Bx] \\ &= 2\text{tr}(APBP) \end{aligned} \quad (1.4.15-8)$$

These results are used in the state estimation for nonlinear systems where nonlinear functions (system dynamics or measurement equation) are approximated by a series expansion of up to second order.

### 1.4.16 Mixture Probability Density Functions

A **mixture pdf** is a **weighted sum of pdfs** with the weights summing up to unity.

A **Gaussian mixture** is a pdf consisting of a weighted sum of Gaussian densities

$$p(x) = \sum_{j=1}^n p_j \mathcal{N}(x; \bar{x}_j, P_j) \quad (1.4.16-1)$$

where, obviously,

$$\sum_{j=1}^n p_j = 1 \quad (1.4.16-2)$$

Denote by  $A_j$  the event that  $x$  is Gaussian distributed with mean  $\bar{x}_j$  and covariance  $P_j$ , that is,

$$A_j \triangleq \{x \sim \mathcal{N}(\bar{x}_j, P_j)\} \quad (1.4.16-3)$$

With  $A_j$ ,  $j = 1, \dots, n$ , mutually exclusive and exhaustive, and

$$P\{A_j\} = p_j \quad (1.4.16-4)$$

one can rewrite (1.4.16-1) with the Total Probability Theorem as

$$p(x) = \sum_{j=1}^n p(x|A_j)P\{A_j\} \quad (1.4.16-5)$$

## The Moments of a Mixture

The mean of a mixture is easily seen to be

$$\bar{x} = \sum_{j=1}^n p_j \bar{x}_j \quad (1.4.16-6)$$

that is, the weighted sum of the means of the component densities regardless of the latter's nature.

The covariance of this mixture is

$$\begin{aligned} E[(x - \bar{x})(x - \bar{x})'] &= \sum_{j=1}^n E[(x - \bar{x})(x - \bar{x})' | A_j] p_j \\ &= \sum E[(x - \bar{x}_j + \bar{x}_j - \bar{x})(x - \bar{x}_j + \bar{x}_j - \bar{x})' | A_j] p_j \\ &= \sum p_j E[(x - \bar{x}_j)(x - \bar{x}_j)' | A_j] + \sum (\bar{x}_j - \bar{x})(\bar{x}_j - \bar{x})' p_j \end{aligned} \quad (1.4.16-7)$$

which can be written as

$$E[(x - \bar{x})(x - \bar{x})'] = \sum_{j=1}^n p_j P_j + \tilde{P} \quad (1.4.16-8)$$

In the above

$$\tilde{P} \triangleq \sum (\bar{x}_j - \bar{x})(\bar{x}_j - \bar{x})' p_j \quad (1.4.16-9)$$

is the *spread of the means term*.

An alternative expression for the above is

$$\begin{aligned} \tilde{P} &= \sum \bar{x}_j \bar{x}'_j p_j - \bar{x} \sum \bar{x}'_j p_j - \sum \bar{x}_j p_j \bar{x}' + \bar{x} \bar{x}' \sum p_j \\ &= \sum p_j \bar{x}_j \bar{x}'_j - \bar{x} \bar{x}' \end{aligned} \quad (1.4.16-10)$$

which yields another form for (1.4.16-8)

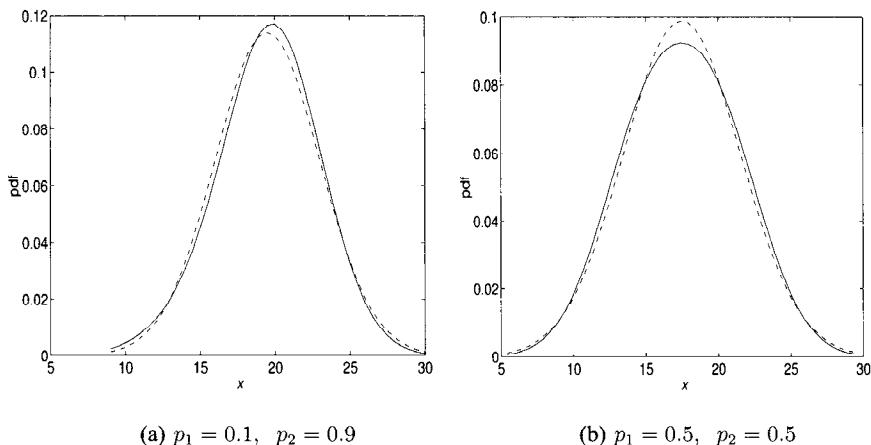
$$E[(x - \bar{x})(x - \bar{x})'] = \sum_{j=1}^n p_j P_j + \sum_{j=1}^n p_j \bar{x}_j \bar{x}'_j - \bar{x} \bar{x}' \quad (1.4.16-11)$$

The spread of the means term  $\tilde{P}$ , defined in (1.4.16-9), is a sum of dyads with positive weightings, and thus it follows from Subsection 1.3.6 that  $\tilde{P}$  is *positive semidefinite*.

Note that (1.4.16-6) and (1.4.16-8), which will be referred to as the *mixture equations*, hold even if the densities in the mixture are *not Gaussian*.

## Approximation of a Mixture

A mixture pdf can be approximated by a single Gaussian pdf with moments equal to those of the mixture, given by (1.4.16-6) and (1.4.16-8) — this is called *moment matching*.



**Figure 1.4.16-1:** The exact Gaussian mixture pdf (solid line) and the corresponding *moment-matched Gaussian* (dashed line) for  $\bar{x}_1 = 15, \bar{x}_2 = 20, \sigma_1^2 = \sigma_2^2 = 10$ .

Figures 1.4.16-1 and 1.4.16-2 present comparisons between the exact pdf of Gaussian mixtures and the corresponding *moment-matched Gaussian* for several values of the parameters entering into the mixture.

In Fig. 1.4.16-1 the difference of the means is 5 while in Fig. 1.4.16-2 it is 10; in both cases the (common) standard deviation of the mixture components is  $\sqrt{10}$ .

The resulting “umbrella” Gaussian appears to be close to the exact pdf of the mixture as long as its components are not too far apart; namely, the distance between the means of the components is up to about *two standard deviations*.

This approximate condition is met for the cases depicted in Fig. 1.4.16-1, where the match appears good, and it is not met for the cases in Fig. 1.4.16-2, where the match does not appear good.

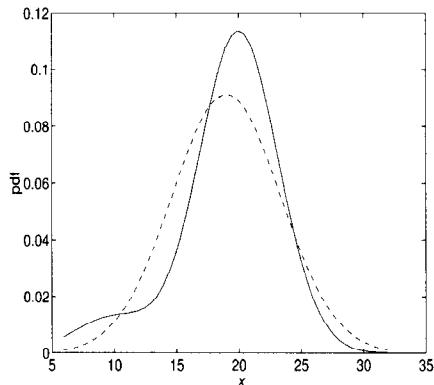
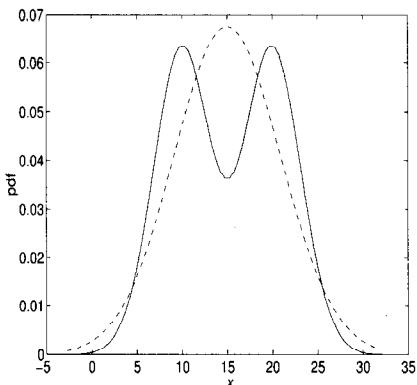
These results are used in obtaining recursive filtering algorithms for systems with hybrid uncertainties — *hybrid systems*. Systems falling into this category are those described by multiple models and/or with measurements of uncertain origin. (See also problem 1-4.)

### 1.4.17 Chi-Square Distributed Random Variables

If the  $n$ -dimensional random vector  $x$  is Gaussian, with mean  $\bar{x}$  and covariance  $P$ , then the (scalar) random variable defined by the quadratic form

$$q = (x - \bar{x})' P^{-1} (x - \bar{x}) \quad (1.4.17-1)$$

can be shown to be the sum of the squares of  $n$  independent zero-mean, unity-variance Gaussian random variables. Such a random variable is said to have a

(a)  $p_1 = 0.1, p_2 = 0.9$ (b)  $p_1 = 0.5, p_2 = 0.5$ 

**Figure 1.4.16-2:** The exact Gaussian mixture pdf (solid line) and the corresponding moment-matched Gaussian (dashed line) for  $\bar{x}_1 = 10, \bar{x}_2 = 20, \sigma_1^2 = \sigma_2^2 = 10$ .

**chi-square distribution with  $n$  degrees of freedom** (the meaning of degrees of freedom is discussed in Subsection 3.6.2).

This can be seen as follows. Let

$$u \triangleq P^{-1/2}(x - \bar{x}) \quad (1.4.17-2)$$

Then  $u$  is Gaussian with

$$E[u] = 0 \quad (1.4.17-3)$$

$$E[uu'] = P^{-1/2}E[x - \bar{x})(x - \bar{x})']P^{-1/2} = P^{-1/2}PP^{-1/2} = I \quad (1.4.17-4)$$

where  $I$  is the identity matrix. Therefore, since the covariance matrix of  $u$  is diagonal, its components are uncorrelated and, in view of the fact that they are jointly Gaussian, they are also independent.

Thus

$$q = u'u = \sum_{i=1}^n u_i^2 \quad (1.4.17-5)$$

where, from (1.4.17-3) and (1.4.17-4),  $u_i$  are now **standard Gaussian random variables** (zero mean and unity variance)

$$u_i \sim \mathcal{N}(0, 1) \quad (1.4.17-6)$$

Therefore  $q$  is chi-square distributed with  $n$  degrees of freedom, which can be written as

$$q \sim \chi_n^2 \quad (1.4.17-7)$$

The mean and variance of the  $\chi_n^2$  random variable  $q$  are

$$E[q] = E\left[\sum_{i=1}^n u_i^2\right] = n \quad (1.4.17-8)$$

$$\begin{aligned}\text{var}(q) &= E\left[\sum_{i=1}^n (u_i^2 - 1)\right]^2 = \sum_{i=1}^n E[(u_i^2 - 1)^2] \\ &= \sum_{i=1}^n (E[u_i^4] - 2E[u_i^2] + 1) = \sum_{i=1}^n (3 - 2 + 1) = 2n \quad (1.4.17-9)\end{aligned}$$

where the fact that the cross-terms are zero mean has been used together with the expression of the fourth moment of a Gaussian variable, as given in (1.4.15-7).

The pdf of  $q$  — the chi-square density with  $n$  degrees of freedom — is

$$p(q) = \frac{1}{2^{\frac{n}{2}} \Gamma\left(\frac{n}{2}\right)} q^{\frac{n-2}{2}} e^{-\frac{q}{2}} \quad q \geq 0 \quad (1.4.17-10)$$

where  $\Gamma$  is the gamma function, with the following useful properties:

$$\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi} \quad \Gamma(1) = 1 \quad \Gamma(m+1) = m\Gamma(m) \quad (1.4.17-11)$$

The square root of  $q$  has the **chi distribution**. The “chi-distributed” random variables with  $n = 2$  and 3 degrees of freedom (and multiplied by an arbitrary constant) have the **Rayleigh distribution** and **Maxwell distribution**, respectively.

Given the *independent* random variables

$$q_1 \sim \chi_{n_1}^2 \quad q_2 \sim \chi_{n_2}^2 \quad (1.4.17-12)$$

then it can be easily shown that their sum

$$q_3 = q_1 + q_2 \quad (1.4.17-13)$$

is chi-square distributed with

$$n_3 = n_1 + n_2 \quad (1.4.17-14)$$

degrees of freedom. Symbolically, one can write

$$\chi_{n_1}^2 + \chi_{n_2}^2 = \chi_{n_1+n_2}^2 \quad (1.4.17-15)$$

The chi-square distribution is often used to check state estimators for “consistency” — that is, whether their actual errors are consistent with the variances calculated by the estimator.

### 1.4.18 Weighted Sum of Chi-Square Random Variables

Consider the following independent and identically distributed random variables

$$x_i \sim \chi_m^2 \quad i = 1, \dots, n \quad (1.4.18-1)$$

with  $m$  degrees of freedom chi-square distribution.

Then the **weighted sum of chi-square variables** with weights  $a_i$ ,

$$y_n \triangleq \sum_{i=1}^n a_i x_i \quad (1.4.18-2)$$

has a distribution which is not chi-square and is very complicated. Its mean is

$$E[y_n] = m \sum_{i=1}^n a_i \quad (1.4.18-3)$$

and, from (1.4.17-9), its variance is

$$\text{var}(y_n) = 2m \sum_{i=1}^n a_i^2 \quad (1.4.18-4)$$

The pdf of (1.4.18-2) can be approximated by **moment matching** — equating its first two moments to those of another random variable. In this case, the latter is chosen as a “scaled” chi-square with a number of degrees of freedom  $n'$ , to be determined together with the scaling factor.

To find a random variable whose moments are matched to those of (1.4.18-2), let

$$v \sim \chi_{n'}^2 \quad (1.4.18-5)$$

Then the mean and variance of a “scaled” version of  $v$ ,

$$w \triangleq cv \quad (1.4.18-6)$$

are  $cn'$  and  $2c^2n'$ , respectively. Equating these two moments to (1.4.18-3) and (1.4.18-4) yields the following equations for  $c$  and  $n'$ :

$$m \sum_{i=1}^n a_i = cn' \quad (1.4.18-7)$$

$$2m \sum_{i=1}^n a_i^2 = 2c^2 n' \quad (1.4.18-8)$$

The solution of these equations yields the scaling factor as

$$c = \frac{\sum_{i=1}^n a_i^2}{\sum_{i=1}^n a_i} \quad (1.4.18-9)$$

and the number of degrees of freedom of  $v$  as

$$n' = \frac{m (\sum_{i=1}^n a_i)^2}{\sum_{i=1}^n a_i^2} \quad (1.4.18-10)$$

Therefore, the distribution of  $y_n$ , defined in (1.4.18-2) is approximately

$$y_n \sim \frac{\sum_{i=1}^n a_i^2}{\sum_{i=1}^n a_i} \chi_{n'}^2 \quad (1.4.18-11)$$

Note that the  $\chi_n^2$  density was defined only for integer  $n$ ; a noninteger  $n$  makes it into a case of the gamma density. In practice, one can interpolate from the chi-square tables.

### Fading Memory Average

Consider the case where the weights in (1.4.18-2) are exponential, that is,

$$a_i = \alpha^{n-i} \quad (1.4.18-12)$$

with  $0 < \alpha < 1$ . Then the weighted sum

$$z_n \triangleq \sum_{i=1}^n \alpha^{n-i} x_i \quad (1.4.18-13)$$

is called the *fading memory sum* of the variables  $x_i$ .

Then, for  $n \gg 1$  one obtains

$$z_n \sim \frac{1}{1 + \alpha} \chi_{n'}^2 \quad (1.4.18-14)$$

and

$$n' = m \frac{1 + \alpha}{1 - \alpha} \quad (1.4.18-15)$$

When (1.4.18-13) is normalized by dividing it with the sum of the coefficients, it becomes the *fading memory average* or *exponentially discounted average*.

These results are used in real-time monitoring of state estimator performance.

### 1.4.19 Random Processes

A scalar random variable is a (real) number  $x$  determined by the outcome  $\omega$  of a random experiment

$$x = x(\omega) \quad (1.4.19-1)$$

A (scalar) *random process* or a *stochastic process* is a *function of time* determined by the outcome of a random experiment

$$x(t) = x(t, \omega) \quad (1.4.19-2)$$

This is a family or **ensemble** of functions of time, in general different for each outcome  $\omega$ .

The **mean** or **ensemble average** of the random process is

$$\bar{x}(t) = E[x(t)] = \int_{-\infty}^{\infty} \xi p_{x(t)}(\xi) d\xi \quad (1.4.19-3)$$

while its **autocorrelation** is defined for a real-valued (scalar) process as

$$R(t_1, t_2) \triangleq E[x(t_1)x(t_2)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \xi \eta p_{x(t_1), x(t_2)}(\xi, \eta) d\xi d\eta \quad (1.4.19-4)$$

The **autocovariance** of this random process is

$$\begin{aligned} V(t_1, t_2) &\triangleq E[[x(t_1) - \bar{x}(t_1)][x(t_2) - \bar{x}(t_2)]] \\ &= R(t_1, t_2) - \bar{x}(t_1)\bar{x}(t_2) \end{aligned} \quad (1.4.19-5)$$

Note that the autocorrelation of a random process is an (unnormalized) *non-central moment* — the mean is not subtracted — while the correlation of two random variables is the normalized joint *central moment* (1.4.6-6). If the process is zero mean, the distinction between central and noncentral moments disappears.

For a **vector-valued random process**, (1.4.19-4) and (1.4.19-5) contain the corresponding outer products.

## Stationarity

A random process whose mean is time invariant and whose autocorrelation is of the form

$$R(t_1, t_2) = R(t_1 - t_2) = R(\tau) \quad (1.4.19-6)$$

where

$$\tau = t_1 - t_2 \quad (1.4.19-7)$$

that is, its first two moments are invariant with respect to a shift of the time axis, is called **wide sense stationary** or, less rigorously, **stationary**.

As can be seen from (1.4.19-4), for a real-valued process the autocorrelation is symmetric in its two time arguments

$$R(t_1, t_2) = R(t_2, t_1) \quad (1.4.19-8)$$

Thus

$$R(\tau) = R(-\tau) \quad (1.4.19-9)$$

The **power spectrum** or **power spectral density** of a *stationary* random process is the Fourier transform (denoted by  $\mathcal{F}$ ) of its autocorrelation

$$S(\omega) = \mathcal{F}\{R(\tau)\} = \int_{-\infty}^{\infty} e^{-j\omega\tau} R(\tau) d\tau \quad (1.4.19-10)$$

where  $\omega$  denotes the (angular) frequency.

**Strict stationarity** requires that all pdfs (rather than only moments up to second order) be invariant with respect to time shift. In practice, wide-sense stationarity is about all one can assume (and hope for).

**Example**

Consider the scalar dynamic system

$$\dot{x}(t) = -ax(t) + n(t) \quad (1.4.19-11)$$

with  $a > 0$ , i.e., a stable system. The input  $n(t)$  is a zero-mean white noise with autocorrelation (same as its autocovariance, since it is zero-mean)

$$E[n(t_1)n(t_2)] = S_0\delta(t_1 - t_2) \quad (1.4.19-12)$$

Assuming that the system in steady state (i.e., the initial condition died out), we obtain

$$x(t) = \int_{-\infty}^t h(t, \tau)n(\tau)d\tau \quad (1.4.19-13)$$

where the impulse response of the system considered is

$$h(t, \tau) = e^{-a(t-\tau)} \quad (1.4.19-14)$$

The mean of the output  $x(t)$  is

$$E[x(t)] = E\left[\int_{-\infty}^t h(t, \tau)n(\tau)d\tau\right] = \int_{-\infty}^t h(t, \tau)E[n(\tau)]d\tau = 0 \quad (1.4.19-15)$$

The autocorrelation of  $x(t)$  is

$$E[x(t_1)x(t_2)] = E\left[\int_{-\infty}^{t_1} h(t_1, \tau_1)n(\tau_1)d\tau_1 \int_{-\infty}^{t_2} h(t_2, \tau_2)n(\tau_2)d\tau_2\right] \quad (1.4.19-16)$$

Assuming  $t_1 < t_2$ , the above can be written as

$$\begin{aligned} E[x(t_1)x(t_2)] &= E\left\{\int_{-\infty}^{t_1} h(t_1, \tau_1)n(\tau_1)d\tau_1 \cdot \left[ \int_{-\infty}^{t_1} h(t_2, \tau_2)n(\tau_2)d\tau_2 + \int_{t_1}^{t_2} h(t_2, \tau_2)n(\tau_2)d\tau_2 \right] \right\} \end{aligned} \quad (1.4.19-17)$$

In view of the fact that the first and last integral above are over nonoverlapping intervals, the expected value of their product will be zero due to the whiteness of  $n(t)$ , whose autocorrelation is zero at any two different times.

Thus one can write

$$\begin{aligned} E[x(t_1)x(t_2)] &= E\left\{\int_{-\infty}^{t_1} e^{-a(t_1-\tau_1)}n(\tau_1)d\tau_1 \int_{-\infty}^{t_1} e^{-a(t_2-\tau_2)}n(\tau_2)d\tau_2\right\} \\ &= e^{a(t_1-t_2)}E\left\{\int_{-\infty}^{t_1} e^{-a(t_1-\tau_1)}n(\tau_1)d\tau_1 \int_{-\infty}^{t_1} e^{-a(t_1-\tau_2)}n(\tau_2)d\tau_2\right\} \\ &= e^{a(t_1-t_2)}E[x(t_1)^2] \end{aligned} \quad (1.4.19-18)$$

If  $t_1 > t_2$  the result would be the same except that they are switched. Therefore,

$$E[x(t_1)x(t_2)] = e^{-a|t_1-t_2|}E[x(t_1)^2] \quad (1.4.19-19)$$

i.e., the autocorrelation decreases exponentially from its peak, which is the (steady-state) mean square value of  $x(t)$  — its instantaneous power.

The mean square value of  $x(t)$  is

$$\begin{aligned} E[x(t)^2] &= E \left\{ \int_{-\infty}^t e^{-a(t-\tau_1)} n(\tau_1) d\tau_1 \int_{-\infty}^t e^{-a(t-\tau_2)} n(\tau_2) d\tau_2 \right\} \\ &= \int_{-\infty}^t \int_{-\infty}^t e^{-a(2t-\tau_1-\tau_2)} E[n(\tau_1)n(\tau_2)] d\tau_1 d\tau_2 \\ &= e^{-2at} \int_{-\infty}^t \int_{-\infty}^t e^{a(\tau_1+\tau_2)} S_0 \delta(\tau_1 - \tau_2) d\tau_1 d\tau_2 \\ &= e^{-2at} S_0 \int_{-\infty}^t e^{2a\tau_1} d\tau_1 = e^{-2at} \frac{S_0}{2a} [e^{2at} - 0] = \frac{S_0}{2a} \end{aligned} \quad (1.4.19-20)$$

Thus

$$R_{xx}(\tau) = E[x(t)x(t+\tau)] = \frac{S_0}{2a} e^{-a|\tau|} \quad (1.4.19-21)$$

The Fourier transform of the above, using the unit step function  $1(\cdot)$  to rewrite the two-sided exponential, yields the spectrum of  $x$  as

$$\begin{aligned} S_{xx}(\omega) &= \mathcal{F}\{R_{xx}(\tau)\} = \frac{S_0}{2a} \mathcal{F}\{e^{a\tau} 1(-\tau) + e^{-a\tau} 1(\tau)\} \\ &= \frac{S_0}{2a} \left[ \frac{1}{a - j\omega} + \frac{1}{a + j\omega} \right] = S_0 \frac{1}{a^2 + \omega^2} \end{aligned} \quad (1.4.19-22)$$

Since this stable system acts as a low-pass filter, its effect on the spectrum of the input (which, being a white noise, has a flat spectrum) is to attenuate the high frequencies.

## Ergodicity

If a stationary random process is **ergodic**, then **time averages** (of some functions of the process) are equal to the corresponding expected values — **ensemble averages** — given by (1.4.19-3), that is,

$$\lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T x(t) dt = \bar{x} \quad (1.4.19-23)$$

Time averages are used in real-time performance monitoring of state estimation filters.

## White Noise

A (not necessarily stationary) random process whose autocovariance is zero for any two different times is called **white noise**. In this case

$$V(t_1, t_2) = \sigma^2(t_1) \delta(t_1 - t_2) = S_0(t_1) \delta(t_1 - t_2) \quad (1.4.19-24)$$

where  $\sigma^2(t_1)$  is its “instantaneous variance” and  $\delta(\cdot)$  is the **Dirac (impulse) delta function**. The above property is the **wide-sense whiteness** as opposed to the **strict-sense whiteness**, which is defined by *independence* rather than *uncorrelatedness* as in (1.4.19-24).

A **stationary zero-mean white** process has the autocorrelation

$$R(\tau) = E[x(t + \tau)x(t)] = S_0\delta(\tau) \quad (1.4.19-25)$$

The impulse function autocorrelation (1.4.19-25) leads to a power spectral density  $S_0$  constant across the frequency spectrum.

It is convenient in some applications to consider a **nonstationary zero-mean white** process  $x(t)$ . The autocorrelation of such a process is

$$E[x(t_1)x(t_2)] = S_0(t_1)\delta(t_1 - t_2) \quad (1.4.19-26)$$

where  $S_0(t)$  is, with some abuse of language, the “time-varying spectral density” or the “instantaneous variance.” The term **instantaneous variance** is in fact *incorrect* in this case since the variance is actually *infinite*.<sup>11</sup>

## Note on Whiteness

In the sequel, when a result requires whiteness, it usually requires strict-sense whiteness. In practice, however, one has only moments (typically up to second order) and thus only wide-sense whiteness can be assumed. Then the result is of practical value only when the strict sense whiteness is replaced by wide-sense whiteness, although in this case the result is only approximately true in theory.

### 1.4.20 Random Walk and the Wiener Process

The **Wiener random process** (or Wiener-Levy or Brownian motion) is a limiting form of the **random walk**: the sum of independent steps of size  $s \rightarrow 0$ , equiprobable in each direction, taken at intervals  $\Delta \rightarrow 0$  such that

$$\frac{s}{\sqrt{\Delta}} \rightarrow \sqrt{\alpha} \quad (1.4.20-1)$$

where  $\alpha$  is a constant. This yields a stochastic process  $w(t)$  with the following pdf [assuming  $w(0) = 0$ ],

$$p[w(t)] = \mathcal{N}[w(t); 0, \alpha t] \quad (1.4.20-2)$$

that is, normal, zero mean, and with variance  $\alpha t$ .

---

<sup>11</sup>Also, the physical dimension of  $\sigma^2$  is the square of the dimension of  $x$ , multiplied by time, since  $\delta$  has as dimension the inverse of time.

Note that the Wiener process is *nonstationary*. It relates to the zero-mean white noise, denoted here as  $n(t)$ , as follows

$$\mathbf{w}(t) = \int_0^t n(\tau) d\tau \quad (1.4.20-3)$$

where

$$E[n(t_1)n(t_2)] = \alpha\delta(t_1 - t_2) \quad (1.4.20-4)$$

Another way of writing (1.4.20-3) is

$$d\mathbf{w}(t) = n(t) dt \quad (1.4.20-5)$$

which shows that the Wiener process is an **independent increment process**. Formally, the white noise is the derivative of the Wiener process; however, this is not rigorous, since the Wiener process is **nowhere differentiable** — its derivative has infinite variance.

The autocorrelation of the Wiener process is

$$E[\mathbf{w}(t_1)\mathbf{w}(t_2)] = \alpha \min(t_1, t_2) \quad (1.4.20-6)$$

The white noise and the Wiener process are used to model unknown inputs (maneuvers) in state estimation/tracking.

## Note

Since the Wiener process is not stationary, it does not have a power spectral density.

### 1.4.21 Markov Processes

**Markov processes** are defined by the following **Markov property**

$$p[x(t)|x(\tau), \tau \leq t_1] = p[x(t)|x(t_1)] \quad \forall t > t_1 \quad (1.4.21-1)$$

that is, the past up to any  $t_1$  is *fully characterized* by the value of the process at  $t_1$ .

An equivalent statement to the above is:

*“The future is independent of the past if the present is known.”*

The Wiener process is Markov. This follows from the fact that it is the integral of white noise

$$\mathbf{w}(t) = \mathbf{w}(t_1) + \int_{t_1}^t n(\tau) d\tau \quad (1.4.21-2)$$

and  $n(\tau)$ ,  $\tau \in [t_1, t]$  is independent of  $\mathbf{w}(t_1)$ .

Furthermore, the state  $x(t)$  of a (possibly time-varying) dynamic system driven by white noise  $n(t)$ ,

$$\dot{x}(t) = f[t, x(t), n(t)] \quad (1.4.21-3)$$

is a Markov process. In general, both  $x(t)$  and  $n(t)$  are vector-valued random processes.

## Markov Processes with Rational Spectra

Given a *linear time-invariant dynamic system excited by a stationary white noise  $n(t)$  with mean zero*

$$\dot{x}(t) = Ax(t) + Bn(t) \quad (1.4.21-4)$$

its state is (in steady state — if the system is stable) a *stationary Markov process* with spectrum (power spectral density)

$$S(\omega) = H(j\omega)QH(j\omega)^* \quad (1.4.21-5)$$

In the above the asterisk denotes complex conjugate transpose and

$$H(j\omega) = (j\omega I - A)^{-1}B \quad (1.4.21-6)$$

is the *transfer function matrix* (from the input to the state) of the system (1.4.21-4) and the autocorrelation of the input (in general, a matrix) is

$$R_n(t_1, t_2) = E[n(t_1)n(t_2)'] = Q\delta(t_1 - t_2) \quad (1.4.21-7)$$

The matrix  $Q$  is sometimes called (nonrigorously) the covariance of  $n(t)$ ; actually it is its *power spectral density*.

Note that (1.4.21-5) is a *rational spectrum* — a ratio of polynomials.

Conversely, one has the following result:

*Every Markov process with a rational spectrum can be represented as a linear time-invariant system excited by white noise.*

The models used in state estimation are Markov processes — linear or nonlinear systems driven by white noise. These systems can be time-invariant or time-varying and the noise can be stationary or nonstationary. If a time-invariant system is driven by a stationary noise that is not white, but has a rational spectrum, then it has to undergo *prewhitening*: It is the output of a subsystem driven by white noise. The transfer function of this subsystem is obtained by *spectral factorization* — decomposition of the output spectrum in the form (1.4.21-5). For more details on spectral factorization, see, e.g., [Poor88].

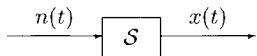
## Prewhitenning

The prewhitening is illustrated in Figs. 1.4.21-1 and 1.4.21-2. In Fig. 1.4.21-1 the system  $\mathcal{S}$  with state  $x(t)$  is driven by the *autocorrelated noise* (also called *colored noise*)  $n(t)$ . Since  $n(t)$  is not white,  $x(t)$  is not a Markov process.

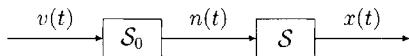
Assume that  $n(t)$  can be represented as the output of a system  $\mathcal{S}_0$  with white noise input  $v(t)$ . With the state of the *prewhitening system* or *shaping filter*  $\mathcal{S}_0$  denoted as  $x_0(t)$ , the augmented state of the composite system shown in Fig. 1.4.21-2,

$$y(t) \triangleq \begin{bmatrix} x(t) \\ x_0(t) \end{bmatrix} \quad (1.4.21-8)$$

is then a Markov process.



**Figure 1.4.21-1:** A system driven by autocorrelated noise.



**Figure 1.4.21-2:** The same system augmented to be driven by white noise.

### Example of Spectral Factorization

Consider the random process  $x(t)$  with the spectrum

$$S_{xx}(\omega) = S_0 \frac{1}{a^2 + \omega^2} \quad (1.4.21-9)$$

This can be factorized as follows. Consider the random process  $x(t)$  with the spectrum

$$S_{xx}(\omega) = \frac{1}{a + j\omega} S_0 \frac{1}{a - j\omega} \quad (1.4.21-10)$$

In the above, one can identify the *causal* transfer function

$$H(\omega) = \frac{1}{a + j\omega} \quad (1.4.21-11)$$

as the transfer function of the system

$$\dot{x}(t) = -ax(t) + n(t) \quad (1.4.21-12)$$

driven by the white noise  $n(t)$ .

If  $n(t)$  is zero-mean with spectrum  $S_0$ , then the above system is the *prewhitening system* for the autocorrelated process  $x(t)$ .

Note that in the course of the factorization, the roles of  $H$  and its complex conjugate  $H^*$  can be switched. It is, however, necessary to choose  $H$  such that it represents a stable and causal system. Namely, the impulse response, given by the inverse Fourier transform, denoted by  $\mathcal{F}^{-1}$ , of the transfer function, i.e.,

$$\mathcal{F}^{-1} \left\{ \frac{1}{a + j\omega} \right\} = e^{-at} 1(t) \quad (1.4.21-13)$$

is **causal**. The impulse response corresponding to the other factor, namely,

$$\mathcal{F}^{-1} \left\{ \frac{1}{a - j\omega} \right\} = e^{at} 1(-t) \quad (1.4.21-14)$$

is **noncausal** (cannot be implemented by a physical system) and thus cannot serve as  $H$ .

### 1.4.22 Random Sequences, Markov Sequences and Markov Chains

A *random sequence*, or a *discrete-time stochastic process*, is a time-indexed sequence of random variables

$$X^k = \{x(j)\}_{j=1}^k \quad k = 1, 2, \dots \quad (1.4.22-1)$$

Similarly to the continuous-time definition of the Markov property, a random sequence is Markov if

$$p[x(k)|X^j] = p[x(k)|x(j)] \quad \forall k > j \quad (1.4.22-2)$$

The (real-valued) *zero-mean sequence*  $v(j)$ ,  $j = 1, \dots$ , is a *discrete-time white noise* (a *white sequence*) if

$$E[v(k)v(j)] = q(k)\delta_{kj} \quad (1.4.22-3)$$

where the *Kronecker delta function*

$$\delta_{kj} = \begin{cases} 1 & \text{if } k = j \\ 0 & \text{if } k \neq j \end{cases} \quad (1.4.22-4)$$

is used and  $q(k)$  denotes its variance. If  $q(k) = q$ , that is, the variance is time-invariant, then this is a *stationary white sequence*.

The sequence with property (1.4.22-3) is actually only uncorrelated, or weakly independent — strong independence requires that (1.4.7-3) be satisfied. However, in practice one does not have pdfs but only moments up to second order, and thus the usual assumption is the *weak independence (uncorrelatedness)* indicated in (1.4.22-3).

The state  $x(k)$  of a dynamic system excited by white noise  $v(k)$

$$x(k+1) = f[k, x(k), v(k)] \quad (1.4.22-5)$$

is a *discrete-time Markov process* or *Markov sequence*. In general, both  $x(k)$  and  $v(k)$  are vector-valued.

The state of a linear dynamic system excited by white Gaussian noise

$$x(k+1) = F(k)x(k) + v(k) \quad (1.4.22-6)$$

is a *Gauss-Markov sequence*.

The reasons for this are as follows: Because of the linearity,  $x(k)$  is Gaussian (assuming the initial condition is Gaussian); and because of the whiteness of the driving (“process”) noise, it is Markov.

A special case of (1.4.22-6), for a scalar  $x$ , is

$$x(k+1) = x(k) + v(k) \quad (1.4.22-7)$$

in which case  $x$  becomes the *integral (sum)* of the white noise sequence terms, and is called a *discrete-time Wiener process*.

## Markov Chains

A **Markov chain** is a special case of a Markov sequence, in which the state space is discrete and finite:

$$x(k) \in \{x_i, i = 1, \dots, n\} \quad (1.4.22-8)$$

Its characterization is given in full by the **transition (jump) probabilities**

$$P\{x(k) = x_j | x(k-1) = x_i\} \triangleq \pi_{ij} \quad i, j = 1, \dots, n \quad (1.4.22-9)$$

and the initial probabilities.

Define the vector

$$\mu(k) \triangleq [\mu_1(k), \dots, \mu_n(k)]' \quad (1.4.22-10)$$

where the components are the probabilities of the chain being in state  $i$

$$\mu_i(k) \triangleq P\{x(k) = x_i\} \quad (1.4.22-11)$$

This vector describes the pmf of the state of the chain.

The evolution in time of (1.4.22-11) is then given by

$$\mu_i(k+1) = \sum_{j=1}^n \pi_{ji} \mu_j(k) \quad i = 1, \dots, n \quad (1.4.22-12)$$

It can be easily shown that the above can be written in vector form with notation (1.4.22-10) as

$$\mu(k+1) = \Pi' \mu(k) \quad (1.4.22-13)$$

where

$$\Pi = [\pi_{ij}] \quad (1.4.22-14)$$

is the **transition matrix of the Markov chain**.

### 1.4.23 The Law of Large Numbers and the Central Limit Theorem

#### The Law of Large Numbers

The **law of large numbers (LLN)** states loosely that the sum of a large number of random variables tends, under some fairly nonrestrictive conditions, to its expected value. One of the versions of the LLN is the following.

Given a stationary sequence of random variables  $x_i, i = 1, \dots$ , with means

$$E[x_i] = \bar{x} \quad (1.4.23-1)$$

and covariances

$$E[(x_i - \bar{x}_i)(x_j - \bar{x}_j)] = \sigma^2 \rho(i-j) \quad (1.4.23-2)$$

where the correlation coefficients are such that

$$\lim_{|i-j|\rightarrow\infty} \rho(i-j) = 0 \quad (1.4.23-3)$$

that is, this sequence is *asymptotically uncorrelated*.

If the correlation coefficients in (1.4.23-3) tend to zero “sufficiently fast” (e.g., exponential decay), then the **sample average**

$$y_n \triangleq \frac{1}{n} \sum_{i=1}^n x_i \quad (1.4.23-4)$$

converges, as  $n \rightarrow \infty$ , in the mean square sense to its expected value

$$\bar{y}_n = \frac{1}{n} \sum_{i=1}^n \bar{x} = \bar{x} \quad (1.4.23-5)$$

In other words, the variance of  $y_n$  in (1.4.23-4) tends to zero as  $n$  increases.

If the stronger condition of uncorrelatedness holds, then the convergence of  $y_n$  given by (1.4.23-4) to  $\bar{y}_n$  given by (1.4.23-5) is quite obvious — it can be easily shown that in this case the mean square value of their difference is  $\sigma^2/n$ .

### The Central Limit Theorem

The **central limit theorem (CLT)** states that if the sequence  $x_i$ ,  $i = 1, \dots$ , consists of *independent* random variables, then under some reasonably mild conditions the pdf of the sum

$$z_n = \frac{1}{\sqrt{n}} \sum_{i=1}^n x_i \quad (1.4.23-6)$$

will tend to a Gaussian pdf as  $n \rightarrow \infty$ .

If these random variables are **independent and identically distributed (i.i.d.)** zero-mean and with variance  $\sigma^2$ , then the  $z_n$  tends to  $\mathcal{N}(0, \sigma^2)$ .

Since  $y_n$  defined in (1.4.23-4) is a scaled version of  $z_n$ , the central limit theorem holds for  $y_n$  as well. Thus, if the random variables  $x_i$  are i.i.d., then for large  $n$ ,

$$y_n \sim \mathcal{N}(\bar{y}_n, \sigma_{y_n}^2) \quad (1.4.23-7)$$

where

$$\bar{y}_n = \bar{x} \quad (1.4.23-8)$$

$$\sigma_{y_n}^2 = \frac{1}{n^2} \sum_{i=1}^n \sigma^2 = \frac{\sigma^2}{n} \quad (1.4.23-9)$$

are the mean and variance of  $y_n$ .

The independence requirement can be weakened — there are versions of the central limit theorem that allow dependence, as long as the random variables become “asymptotically independent,” in a manner somewhat similar to the asymptotic uncorrelatedness property (1.4.23-3).

## Note on the CLT

The CLT has a very important role in characterizing many real-world sources of uncertainty: It is used as the justification/excuse to make the omnipresent Gaussian assumption. For example, the thermal noise in electronic devices, as the sum of many “small contributions,” is indeed close to Gaussian.

## 1.5 BRIEF REVIEW OF STATISTICS

### 1.5.1 Hypothesis Testing

Consider the *hypothesis testing* problem between the following “simple” hypotheses — each is defined by a point (a certain value of a parameter).

The *null hypothesis* is

$$H_0 : \theta = \theta_0 \quad (1.5.1-1)$$

while the *alternate hypothesis* is

$$H_1 : \theta = \theta_1 \quad (1.5.1-2)$$

where  $\theta$  is a certain parameter whose value is equal to a certain value  $\theta_0$  ( $\theta_1$ ) under  $H_0$  ( $H_1$ ).

The *type I error* probability is defined as

$$P_{e_I} \triangleq P\{\text{“}H_1\text{”}|H_0\} \triangleq P\{\text{accept } H_1 | H_0 \text{ true}\} \quad (1.5.1-3)$$

while the *type II error* probability is

$$P_{e_{II}} \triangleq P\{\text{“}H_0\text{”}|H_1\} \triangleq P\{\text{accept } H_0 | H_1 \text{ true}\} \quad (1.5.1-4)$$

In signal detection, if  $H_0$  stands for “signal equal to zero” (i.e., absent) and  $H_1$  stands for “signal present,” then the type I error is a *false alarm* while the type II error is a *miss*.

The *power of the test* is

$$\pi \triangleq P\{\text{“}H_1\text{”}|H_1\} = 1 - P_{e_{II}} \quad (1.5.1-5)$$

and it measures the test’s capability to discern  $H_1$  when it is true — the *detection probability*.

The decision as to which hypothesis to accept is made based on a set of observations,  $Z$ , whose pdfs conditioned on  $H_0$  and  $H_1$  are known.

According to the *Neyman-Pearson Lemma*, the optimal decision, in the sense of *minimizing the probability of type II error* (or maximizing the power of the test), *subject to a given (maximum) probability of type I error* is as follows.

The test, based on the *likelihood ratio*, is

$$\Lambda(H_1, H_0) = \frac{p(Z|H_1)}{p(Z|H_0)} \stackrel{\text{"}H_1\text{"}}{\underset{\text{"}H_0\text{"}}{\gtrless}} \Lambda_0 \quad (1.5.1-6)$$

that is, “accept  $H_1$ ” if  $\Lambda$  exceeds the threshold  $\Lambda_0$  and “accept  $H_0$ ” if  $\Lambda$  is below this threshold. The threshold  $\Lambda_0$  is such that

$$P\{\Lambda(H_1, H_0) > \Lambda_0 | H_0\} = P_{e_I} \quad (1.5.1-7)$$

The pdf of the observations,  $p(Z|H_j)$ , is called the *likelihood function* of  $H_j$ .

### Example

Consider the test between the hypotheses (1.5.1-1) and (1.5.1-2) with  $\theta_0 = 0$  and  $\theta_1 > 0$ , based on the single observation

$$z \sim \mathcal{N}(\theta, \sigma^2) \quad (1.5.1-8)$$

that is, we are testing whether the mean of  $z$  is zero or positive.

The likelihood ratio is

$$\Lambda(H_1, H_0) = \frac{p(z|H_1)}{p(z|H_0)} = e^{-\frac{(z-\theta_1)^2 - z^2}{2\sigma^2}} = e^{\frac{2z\theta_1 - \theta_1^2}{2\sigma^2}} \quad (1.5.1-9)$$

Instead of comparing (1.5.1-9) to a threshold, it is convenient to take its logarithm, incorporate the various constants into the threshold, and then determine it.

Since  $\theta_1 > 0$ , comparing (1.5.1-9) to a threshold is equivalent to

$$z \stackrel{\text{"}H_1\text{"}}{\underset{\text{"}H_0\text{"}}{\gtrless}} \lambda_0 \quad (1.5.1-10)$$

where  $\lambda_0$  is the new threshold to be determined.

In view of (1.5.1-8), the pdf of the observation under  $H_0$  is

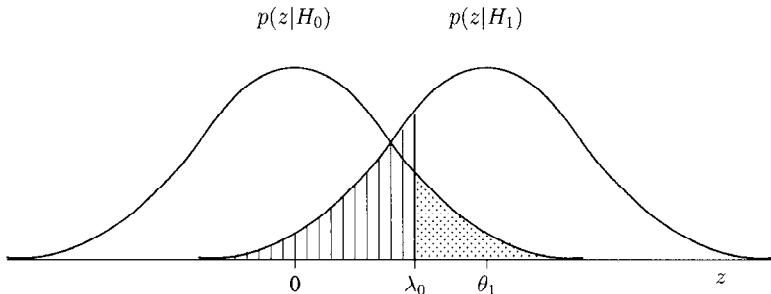
$$p(z|H_0) = \mathcal{N}(z; 0, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{z^2}{2\sigma^2}} \quad (1.5.1-11)$$

The threshold  $\lambda_0$  for *false alarm probability*  $P_{FA} = \alpha$  follows from

$$P_{FA} \triangleq P_{e_I} = P\{z > \lambda_0 | H_0\} = \int_{\lambda_0}^{\infty} \mathcal{N}(z; 0, \sigma^2) dz = \alpha \quad (1.5.1-12)$$

The *tail mass* or *tail probability*  $\alpha$  from (1.5.1-12) is shaded in Fig. 1.5.1-1.

Under  $H_0$  there is only a small probability  $\alpha$  that  $z$  falls in the shaded tail region of the likelihood function of  $H_0$ ; if it does fall there, then it is deemed



**Figure 1.5.1-1:** Test between two simple hypotheses.

unlikely that  $H_0$  is true, and it is rejected. In this case it is said that the *level of significance* of  $H_0$  is low:  $\alpha$  or less.<sup>12</sup> The latter holds if  $z = z_1 > \lambda_0$  because such an observation belongs to an even smaller tail mass.

The type II error probability — the *miss probability* — of the test (1.5.1-10) is

$$P_{e_{II}} \triangleq P\{z < \lambda_0 | H_1\} = \int_{-\infty}^{\lambda_0} \mathcal{N}(z; \theta_1, \sigma^2) dz \quad (1.5.1-13)$$

This corresponds to the striped area of Fig. 1.5.1-1.

The power of this test — the *probability of detection* — also denoted as  $P_D$  is

$$P_D \triangleq \pi = P\{z > \lambda_0 | H_1\} = \int_{\lambda_0}^{\infty} \mathcal{N}(z; \theta_1, \sigma^2) dz = 1 - P_{e_{II}} \quad (1.5.1-14)$$

and, obviously, it depends on  $\theta_1$  — the farther it is from 0, the more powerful the test will be.

### Example

For  $\alpha = 5\%$ , the threshold is obtained (see Subsection 1.5.4) based on the one-sided tail of the Gaussian pdf as  $\lambda_0 = 1.64\sigma$ . Table 1.5.1-1 shows the power of this test as a function of the separation between the two means (in units of  $\sigma$ , the common standard deviation for both hypotheses).

## 1.5.2 Confidence Regions and Significance

Assume one desires to test the null hypothesis

$$H_0 : \theta = 0 \quad (1.5.2-1)$$

<sup>12</sup>Equivalently, one can say that the *confidence level* for  $H_1$  is high ( $1 - \alpha$  or more).

**Table 1.5.1-1:** Power of the test between simple hypotheses as a function of their separation (threshold set for  $\alpha = 5\%$ ).

$\theta_1/\sigma$	2	3	4	5
Power $\pi$	0.641	0.913	0.991	0.9996

versus the **one-sided alternative hypothesis**

$$H'_1 : \theta > 0 \quad (1.5.2-2)$$

based upon a set of observations

$$Z = \{z_i, i = 1, \dots, n\} \quad (1.5.2-3)$$

where

$$z_i = \theta + w_i \quad (1.5.2-4)$$

and the “noises”  $w_i$  are independent and identically distributed:

$$w_i \sim \mathcal{N}(0, \sigma^2) \quad (1.5.2-5)$$

Note that  $H'_1$  is a **composite hypothesis** — it is defined by more than one point (an interval in this case).

The test is subject to

$$P_{e_I} = \alpha \quad (1.5.2-6)$$

The likelihood ratio for hypothesis  $H'_1$  versus  $H_0$  is

$$\Lambda(H'_1, H_0) = \frac{p(Z|\theta)}{p(Z|\theta=0)} = e^{\frac{1}{2\sigma^2} \sum_{i=1}^n [z_i^2 - (z_i - \theta)^2]} = e^{\frac{1}{2\sigma^2} \sum_{i=1}^n (2z_i\theta - \theta^2)} \quad (1.5.2-7)$$

The comparison of the above to a threshold is equivalent to comparing the sample mean of the observations to another threshold:

$$\bar{z} \triangleq \frac{1}{n} \sum_{i=1}^n z_i \underset{\text{"}H_0\text{"}}{\underset{\text{"}H'_1\text{"}}{\gtrless}} \lambda_1 \quad (1.5.2-8)$$

Equation (1.5.2-8) follows by taking the logarithm of (1.5.2-7) and lumping  $\theta$  and  $\sigma$  into the (yet undetermined) threshold; the division by  $n$  is done for the convenience of normalization — this yields  $\bar{z}$  in the same range regardless of the number of samples  $n$ .

For this problem  $\bar{z}$  is the **test statistic** — the function of the observations used in the test.

The threshold  $\lambda_1$  is obtained by noting that

$$p(\bar{z}|H_0) = \mathcal{N}(\bar{z}; 0, \frac{\sigma^2}{n}) \quad (1.5.2-9)$$

and putting the condition

$$P\{\text{“}H'_1\text{”}|H_0\} = P\{\bar{z} > \lambda_1|H_0\} = 1 - \int_{-\infty}^{\lambda_1} \mathcal{N}(\bar{z}; 0, \frac{\sigma^2}{n}) d\bar{z} = \alpha \quad (1.5.2-10)$$

If the one-sided alternate hypothesis (1.5.2-2) is replaced by the **two-sided alternate hypothesis**, denoted as  $H_1$ ,

$$H_1 : \theta \neq 0 \quad (1.5.2-11)$$

then it can be shown that (1.5.2-8) is to be replaced by

$$|\bar{z}| \underset{\substack{\text{“}H_1\text{”} \\ \text{“}H_0\text{”}}}{\gtrless} \lambda \quad (1.5.2-12)$$

The threshold  $\lambda$  is obtained by putting the condition

$$P\{\text{“}H_1\text{”}|H_0\} = P\{|\bar{z}| > \lambda|H_0\} = 1 - \int_{-\lambda}^{\lambda} \mathcal{N}(\bar{z}; 0, \frac{\sigma^2}{n}) d\bar{z} = \alpha \quad (1.5.2-13)$$

In other words,  $\lambda$  is such that the **acceptance region** for  $H_0$ , which is the interval  $[-\lambda, \lambda]$ , contains  $1 - \alpha$  probability mass for the pdf (1.5.2-9).

For  $\alpha = 0.05$ , one obtains from tables of the normal distribution (see Sub-section 1.5.4)

$$\lambda = 1.96 \frac{\sigma}{\sqrt{n}} \quad (1.5.2-14)$$

The power of the test depends on the specific value of  $\theta$  under the alternate hypothesis  $H_1$ .

The sample mean  $\bar{z}$  is also the **maximum likelihood estimate** (and the **least squares estimate**) of the unknown parameter  $\theta$ , i.e.,

$$\bar{z} = \arg \max_{\theta} p(Z|\theta) = \arg \min_{\theta} \sum_{i=1}^n (z_i - \theta)^2 \quad (1.5.2-15)$$

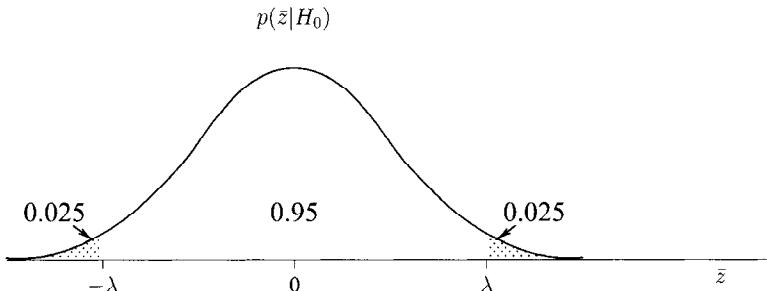
since  $p(Z|\theta)$  is the likelihood function of the parameter  $\theta$ .

Figure 1.5.2-1 illustrates the acceptance region for  $H_0$  — the **two-sided 95% probability region** for  $\bar{z}$  under hypothesis  $H_0$  — the interval  $[-\lambda, \lambda]$ . The shaded areas represent the tails on the two sides of  $p(\bar{z}|H_0)$ , whose total probability mass is  $\alpha = 0.05$ ; this is the region of rejection of  $H_0$  (acceptance of  $H_1$ ).

Since the difference between the sample mean  $\bar{z}$  and the true mean  $\theta$  (whatever its value) is

$$\bar{z} - \theta \sim \mathcal{N}(0, \frac{\sigma^2}{n}) \quad (1.5.2-16)$$

one can say that the true mean lies within the interval  $[\bar{z} - \lambda, \bar{z} + \lambda]$  with “confidence”  $1 - \alpha$  (since  $\theta$  is not a random variable, no probabilistic statement about it can be made). In view of this, such an interval is called the **confidence region** for  $\theta$ .



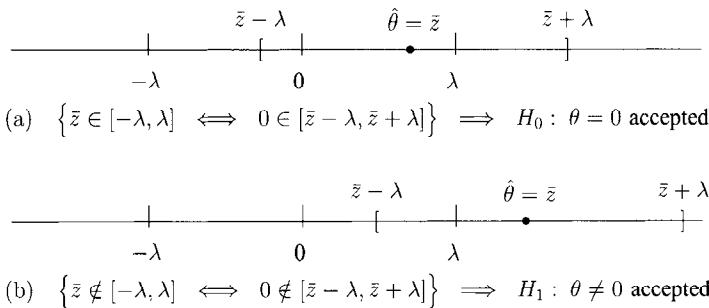
**Figure 1.5.2-1:** The test between hypotheses  $H_0$  and  $H_1$ .

Another way of interpreting test (1.5.2-12) is the following: If the null hypothesis value  $\theta = 0$  falls within the confidence region  $[\bar{z} - \lambda, \bar{z} + \lambda]$ , then  $H_0$  is accepted. This is equivalent to  $\bar{z}$  falling within the acceptance region  $[-\lambda, \lambda]$ .

If the estimate  $\hat{\theta} = \bar{z}$  of  $\theta$  falls outside the interval  $[-\lambda, \lambda]$ , then it is said to be a **significant estimate** — hypothesis  $H_1$  is accepted because  $H_0$  is **insignificant**.

Note that while  $\bar{z}$  is the **point estimate** of  $\theta$ , the confidence region around  $\bar{z}$  can be seen as an **interval estimate** of the parameter  $\theta$ .

Figure 1.5.2-2 illustrates these concepts.



**Figure 1.5.2-2:** Parameter estimate. (a)  $\hat{\theta}$  insignificant; (b)  $\hat{\theta}$  significant.

Thus the estimate of a parameter that might be zero is accepted as *significant* (at a “level” of 5% for the null hypothesis) if

$$\frac{|\hat{\theta}|}{\sigma_{\hat{\theta}}} > 1.96 \quad (1.5.2-17)$$

where the threshold has been determined under the normal assumption and

$$\sigma_{\hat{\theta}} = \frac{\sigma}{\sqrt{n}} \quad (1.5.2-18)$$

is the standard deviation of the estimate, or its **standard error**.

Defining the **estimation error** as

$$\tilde{\theta} \triangleq \hat{\theta} - \theta \quad (1.5.2-19)$$

one can see that

$$\begin{aligned} p(\hat{\theta}|\theta) &= \mathcal{N}(\hat{\theta}; \theta, \sigma_{\hat{\theta}}^2) = \mathcal{N}(\hat{\theta} - \theta; 0, \sigma_{\tilde{\theta}}^2) \\ &= \mathcal{N}(\tilde{\theta}; 0, \sigma_{\tilde{\theta}}^2) \triangleq p(\tilde{\theta}) = \mathcal{N}(\tilde{\theta}; 0, \sigma_{\tilde{\theta}}^2) \end{aligned} \quad (1.5.2-20)$$

Therefore (1.5.2-18), which is the **standard deviation of the estimate** (about the true mean),  $\sigma_{\hat{\theta}}$ , is the same as  $\sigma_{\tilde{\theta}}$ , the **standard deviation of the estimation error** (which is zero mean), that is,

$$\sigma_{\hat{\theta}} = \sigma_{\tilde{\theta}} \quad (1.5.2-21)$$

The (statistical) significance of parameter estimates is used in choosing the order of models to describe system equations (e.g., for target motion). (See also problem 1-2.)

### Example

A parameter  $\theta$  is estimated based on  $n$  measurements, each with an additive error with standard deviation  $\sigma = 10$ . Assume the estimate obtained from  $n = 10$  measurements is  $\hat{\theta} = 2$ . The significance statistic (at a level of 5%) is in this case

$$\frac{|\hat{\theta}|}{\sigma_{\hat{\theta}}} = \frac{|\hat{\theta}|}{\sigma/\sqrt{n}} = \frac{2}{10/\sqrt{10}} = 0.63 \ll 1.96 \quad (1.5.2-22)$$

i.e., based on these  $n = 10$  measurements, this estimate is clearly not statistically significant — the true value of the parameter can be assumed as zero. The threshold above corresponds to 5% level of significance (5% probability that the estimate is “due to noise only,” or 95% confidence that it is nonzero).

Assume next that the observations are repeated until one has  $n = 100$  measurements and that the estimate is still  $\hat{\theta} = 2$ . The significance statistic is now

$$\frac{|\hat{\theta}|}{\sigma/\sqrt{n}} = \frac{2}{10/\sqrt{100}} = 2 > 1.96 \quad (1.5.2-23)$$

i.e., based on these  $n = 100$  measurements, this estimate is statistically significant, even though only marginally.

Note that it is somewhat arbitrary to use a 5% level of significance level for the threshold for acceptance/rejection of the null hypothesis. A marginally significant parameter can become not significant if the above level is lowered. However, while there is some subjectivity in the selection of the level for the test, this only affects marginal cases. For example, if  $\hat{\theta} = 12$ , then this value would be accepted even from  $n = 10$  measurements since the test statistic is

$$\frac{|\hat{\theta}|}{\sigma/\sqrt{n}} = \frac{12}{10/\sqrt{10}} = 3.8 \quad (1.5.2-24)$$

i.e., large enough to exceed even the 1% level threshold, which is 3.38.

### 1.5.3 Monte Carlo Runs and Comparison of Algorithms

The performance of an estimation (or control) algorithm is usually quantified by the expected value of a “cost” function

$$J = E[C] \quad (1.5.3-1)$$

If  $C$  is, for instance, the squared error in the estimation of a certain variable, then  $J$  is the corresponding mean square error.

In many situations, the performance of an algorithm of interest cannot be evaluated analytically.<sup>13</sup> In such a case, *Monte Carlo simulations (runs)* are made to obtain an estimate of  $J$  from a sample average of *independent* realizations  $C_i$ ,  $i = 1, \dots, N$ , of the cost  $C$ . The larger the number of such runs, the smaller is the variability (error) of the resulting estimate. Also, a larger number of runs increases the power of the hypothesis testing used in comparing different algorithms.

The estimate of the performance from  $N$  independent runs is the *sample average* (or *sample mean*) of the  $N$  realizations of the cost

$$\bar{C} = \frac{1}{N} \sum_{i=1}^N C_i \quad (1.5.3-2)$$

with the associated standard deviation — the *standard error*

$$\sigma_{\bar{C}} = \sqrt{\frac{1}{N^2} \sum_{i=1}^N (C_i - \bar{C})^2} \quad (1.5.3-3)$$

The above follows from the fact that the variance of the sample mean  $\bar{C}$  is the variance of  $C$  divided by  $N$ ; since this variance is not known, it is replaced by the sample variance, which has another  $N$  in the denominator. This is the reason for having  $N^2$  in the denominator of (1.5.3-3).

---

<sup>13</sup>Those who can, do. Those who cannot, simulate. Those who cannot simulate, supervise.

These can be used to obtain a confidence region for the performance assuming that its distribution is, in view of the central limit theorem, approximately normal. Since the pdf of (1.5.3-2) is in general skewed, confidence regions based on the normal assumption might not be accurate unless  $N$  is very large — possibly of the order of thousands.

## Algorithm Comparison as a Hypothesis Testing Problem

When two algorithms are compared based on simulations, this can be formulated as a hypothesis testing problem as follows.

Assume one has the sample mean of the performance (e.g., MSE, or some other cost to be minimized) for algorithm  $j$  from  $N$  independent runs

$$\bar{C}^{(j)} = \frac{1}{N} \sum_{i=1}^N C_i^{(j)} \quad j = 1, 2 \quad (1.5.3-4)$$

where  $C_i^{(j)}$  is the performance of algorithm  $j$  in run  $i$  and  $C_i^{(j)}$  is independent of  $C_k^{(j)}$ ,  $\forall k \neq i$ . Each pair of runs  $i$  uses the *same random variables for the two algorithms*  $j = 1, 2$ . As shown in problem 1-3, it is *beneficial* to use the same random variables for both algorithms in the same run.

## Some Simple Comparison Techniques

Assume that from the simulations one has

$$\bar{C}^{(1)} < \bar{C}^{(2)} \quad (1.5.3-5)$$

This does not necessarily imply that algorithm 1 is better than algorithm 2. Any such statement must be qualified by a probability  $\alpha$  of error of type I.

Comparison technique #0, based on (1.5.3-5) without any statistical analysis of the individual run outcomes, is clearly *naive*.

Comparison technique #1 — the simplest (but still simplistic) statistical approach — consists of the following:

1. Calculate confidence regions around the two sample means (based on (1.5.3-2) and (1.5.3-3)).
2. If they do not overlap, declare that the algorithm with the smaller sample mean is the superior one.

The last step above implicitly assumes the two sample means to be uncorrelated. Note that this is *incorrect*, since each pair of runs uses the same random numbers.

## The Optimal Comparison Technique

Since  $C_i^{(1)}$  is correlated with  $C_i^{(2)}$  (because they use the same random variables), the optimal test, which will be based on *independent samples*, is as follows.

The hypothesis testing problem is

$$H_0 : \Delta = J^{(2)} - J^{(1)} \leq 0 \quad (\text{algorithm 1 not better than 2}) \quad (1.5.3-6)$$

versus

$$H_1 : \Delta = J^{(2)} - J^{(1)} > 0 \quad (\text{algorithm 1 better than 2}) \quad (1.5.3-7)$$

subject to

$$P\{\text{accept } H_1 | H_0 \text{ true}\} = \alpha \quad (\text{level of significance (of hypothesis } H_0)) \quad (1.5.3-8)$$

where

$$J^{(j)} = E[C^{(j)}] \quad j = 1, 2 \quad (1.5.3-9)$$

are the *true expected values of the cost functions (true performance)*.

The decision whether  $H_1$  (“1” better than “2”) should be accepted is made based upon the *sample performance differences*

$$\boxed{\Delta_i = C_i^{(2)} - C_i^{(1)}} \quad (1.5.3-10)$$

Note that  $\Delta_i$  is independent of  $\Delta_k$ ,  $\forall k \neq i$ .

Then,  $H_1$  is accepted if

$$\boxed{\mu \triangleq \frac{\bar{\Delta}}{\sigma_{\bar{\Delta}}} > \mu_0} \quad (1.5.3-11)$$

where

$$\boxed{\bar{\Delta} = \frac{1}{N} \sum_{i=1}^N \Delta_i} \quad (1.5.3-12)$$

and

$$\boxed{\sigma_{\bar{\Delta}} = \sqrt{\frac{1}{N^2} \sum_{i=1}^N (\Delta_i - \bar{\Delta})^2}} \quad (1.5.3-13)$$

are the *sample mean* of the differences (1.5.3-10) and the *standard error* of this sample mean, respectively.

Assuming the error in  $\bar{\Delta}$  to be normal, the threshold  $\mu_0$  is based on the *upper tail* of the normal density:  $\mu_0 = 1.64$  for  $\alpha = 5\%$ ,  $\mu_0 = 2.33$  for  $\alpha = 1\%$ , and so on (see Subsection 1.5.4). This follows from the fact that the test is for *positive mean ( $H_1$ ) versus zero or negative mean* and  $H_1$  is accepted if and only if the sample mean (1.5.3-12) is positive and statistically significant (1.5.3-11);  $\alpha$  is the probability of accepting  $H_1$  (positive mean) when the true mean is zero.

This procedure is summarized below:

$C_1^{(1)}$	$C_1^{(2)}$	$\Delta_1$
$\vdots$	$\vdots$	$\vdots$
$C_N^{(1)}$	$C_N^{(2)}$	$\Delta_N$
$\bar{C}^{(1)}$	$\bar{C}^{(2)}$	$\bar{\Delta}, \sigma_{\bar{\Delta}}$

## Remarks

The optimal algorithm comparison is based on the significance test of the *sample mean of the differences* (the mean of the terms in the last column above) instead of the naive comparison of the performance estimates from the first two columns (technique #0) or using two confidence regions around these estimates (technique #1).

The applicability of the CLT is much more realistic on the average difference (1.5.3-12) than on the average outcomes (1.5.3-4) since the differences are much less skewed (and independent from run to run).

The fact that the same random numbers are used for the two algorithms leads to a *positive correlation* between the two sample means, and the optimal test presented takes advantage of this. (See also problem 1-3.)

Section 11.5 presents a major example of the application of this optimal comparison technique.

### 1.5.4 Tables of the Chi-Square and Gaussian Distributions

Table 1.5.4-1 presents the points  $x$  on the chi-square distribution for a given *upper tail probability*

$$Q = P\{y > x\} \quad (1.5.4-1)$$

where

$$y \sim \chi_n^2 \quad (1.5.4-2)$$

and  $n$  is the number of degrees of freedom. This tabulated function is also known as the *complementary distribution*.

An alternative way of writing (1.5.4-1) is

$$x(1 - Q) \triangleq \chi_n^2(1 - Q) \quad (1.5.4-3)$$

which indicates that to the left of the point  $x$  the probability mass is  $1 - Q$ . This is the  $100(1 - Q)$  *percentile point*.

The 95% probability region for a  $\chi_n^2$  variable can be taken as the *one-sided probability region* (cutting off the 5% *upper tail*)

$$[0, \chi_2^2(0.95)] = [0, 5.99] \quad (1.5.4-4)$$

or the **two-sided probability region** (cutting off both 2.5% tails)

$$[\chi_2^2(0.025), \chi_2^2(0.975)] = [0.05, 7.38] \quad (1.5.4-5)$$

For a  $\chi_{100}^2$  variable, the two-sided 95% probability region is

$$[\chi_{100}^2(0.025), \chi_{100}^2(0.975)] = [74, 130] \quad (1.5.4-6)$$

Note the skewedness of the chi-square distribution: The above two-sided regions are not symmetric about the corresponding means (2 and 100, respectively, in the above two equations).

For degrees of freedom above 100, the following approximation of the points on the chi-square distribution can be used:

$$\chi_n^2(1 - Q) = \frac{1}{2} [\mathcal{G}(1 - Q) + \sqrt{2n - 1}]^2 \quad (1.5.4-7)$$

where  $\mathcal{G}$  is defined in (1.5.4-11). For example,  $\chi_{400}^2(0.025) = 346$ ,  $\chi_{400}^2(0.975) = 457$ .

The last line of Table 1.5.4-1 shows the points  $x$  on the **standard** (zero mean and unity variance) **normal (Gaussian)** distribution for the same tail probabilities. In this case

$$y \sim \mathcal{N}(0, 1) \quad (1.5.4-8)$$

and, with

$$Q = P\{y > x\} \quad (1.5.4-9)$$

these points will be denoted as

$$x(1 - Q) \triangleq \mathcal{G}(1 - Q) \quad (1.5.4-10)$$

Thus, with this notation the 95% **two-sided probability region** for an  $\mathcal{N}(0, 1)$  random variable is

$$[\mathcal{G}(0.025), \mathcal{G}(0.975)] = [-1.96, 1.96] \quad (1.5.4-11)$$

In terms of the cumulative distribution function (cdf) of a standard Gaussian random variable

$$\begin{aligned} P_G(x) &\triangleq P\{y \leq x\} \\ &= \int_{-\infty}^x \mathcal{N}(y; 0, 1) dy \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{y^2}{2}} dy \end{aligned} \quad (1.5.4-12)$$

one can write (1.5.4-11) as

$$P_G(1.96) = 0.975 \quad (1.5.4-13)$$

$$P_G(-1.96) = 0.025 \quad (1.5.4-14)$$

Note that  $P_G$  defined in (1.5.4-12) is the inverse function of  $\mathcal{G}$  defined in (1.5.4-10), that is,  $\mathcal{G}$  is the inverse cdf.

The 95% **one-sided probability region** for a standard Gaussian random variable is given by its 95% point:

$$(-\infty, \mathcal{G}(0.95)] = (-\infty, 1.64] \quad (1.5.4-15)$$

**Table 1.5.4-1:** Tail probabilities of the chi-square and normal densities.

$n \setminus Q$	0.99	0.975	0.95	0.90	0.75	0.5	0.25	0.10	0.05	0.025	0.01	5E-3	1E-3
1	2E-4	.001	.003	.016	.102	.455	1.32	2.71	3.84	5.02	6.63	7.88	10.8
2	.020	.051	.103	.211	.575	1.39	2.77	4.61	5.99	7.38	9.21	10.6	13.8
3	.115	.216	.352	.584	1.21	2.37	4.11	6.25	7.81	9.35	11.3	12.8	16.3
4	.297	.484	.711	1.06	1.92	3.36	5.39	7.78	9.49	11.1	13.3	14.9	18.5
5	.554	.831	1.15	1.61	2.67	4.35	6.63	9.24	11.1	12.8	15.1	16.7	20.5
6	.872	1.24	1.64	2.20	3.35	5.35	7.84	10.6	12.6	14.4	16.8	18.5	22.5
7	1.24	1.69	2.17	2.83	4.25	6.35	9.04	12.0	14.1	16.1	18.5	20.3	24.3
8	1.65	2.18	2.73	3.49	5.07	7.34	10.2	13.4	15.5	17.5	20.1	22.0	26.1
9	2.09	2.70	3.33	4.17	5.90	8.34	11.4	14.7	17.0	19.0	21.7	23.6	27.9
10	2.56	3.25	3.94	4.87	6.74	9.34	12.5	16.0	18.3	20.5	23.2	25.2	29.6
11	3.05	3.82	4.57	5.58	7.58	10.3	13.7	17.3	19.7	22.0	24.7	26.8	31.3
12	3.57	4.40	5.23	6.30	8.44	11.3	14.8	18.5	21.0	23.3	26.2	28.3	32.9
13	4.11	5.01	5.90	7.04	9.30	12.3	16.0	19.8	22.4	24.7	27.7	29.8	34.5
14	4.66	5.63	6.57	7.79	10.2	13.3	17.1	21.1	23.7	26.1	29.1	31.3	36.1
15	5.23	6.26	7.26	8.55	11.0	14.3	18.2	22.3	25.0	27.5	30.6	32.8	37.7
16	5.81	6.91	7.96	9.31	11.9	15.3	19.4	23.5	26.3	28.8	32.0	34.3	39.3
17	6.41	7.56	8.67	10.1	12.8	16.3	20.5	24.8	27.6	30.2	33.4	35.7	40.8
18	7.01	8.23	9.40	10.9	13.7	17.3	21.6	26.0	28.9	31.5	34.8	37.2	42.3
19	7.63	8.91	10.1	11.7	14.6	18.3	22.7	27.2	30.1	32.9	36.2	38.6	43.8
20	8.26	9.60	10.9	12.4	15.5	19.3	23.8	28.4	31.4	34.2	37.6	40.0	45.3
25	11.5	13.1	14.6	16.5	19.9	24.3	29.3	34.4	37.7	40.6	44.3	46.9	52.6
30	15.0	16.8	18.5	20.6	24.5	29.3	34.8	40.3	43.8	47.0	50.9	53.7	59.7
40	22.2	24.4	26.5	29.1	33.7	39.3	45.6	51.8	55.8	59.3	63.7	66.8	73.4
50	29.7	32.4	34.8	37.7	43.0	49.3	56.3	63.2	67.5	71.4	76.2	79.5	86.7
60	37.5	40.5	43.2	46.5	52.3	59.3	67.0	74.4	79.1	83.3	88.4	92.0	99.6
70	45.4	48.8	51.7	55.3	61.7	69.3	77.6	85.5	90.5	95.0	100	104	112
80	53.5	57.2	60.4	64.2	71.1	79.3	88.1	96.6	102	107	112	116	125
90	61.8	65.6	69.1	73.3	80.6	89.3	98.6	108	113	118	124	128	137
100	70.1	74.2	77.9	82.4	90.1	99.3	109	118	124	130	136	140	149
150	112	118	123	128	138	149	161	173	180	186	193	198	209
200	157	163	168	175	186	199	213	226	234	241	249	255	268
300	246	254	261	269	283	299	316	332	341	350	360	367	381
400	337	346	355	364	381	399	419	437	448	457	469	478	495
500	429	440	449	460	478	499	521	541	553	563	576	585	603
800	710	724	735	749	773	799	827	852	867	880	896	907	929
$\mathcal{G}()$	-2.33	-1.96	-1.64	-1.28	-0.675	0	0.675	1.28	1.64	1.96	2.33	2.58	3.09

## Some Distribution-free Probability Regions

The *Chebyshev inequality* provides a *distribution-free probability region*. For a random variable  $x$  with mean  $\mu$  and variance  $\sigma^2$ , one has, for any  $\epsilon > 0$ ,

$$P\{|x - \mu| \geq \epsilon\} \leq \frac{\sigma^2}{\epsilon^2} \quad (1.5.4-16)$$

regardless of the distribution of  $x$ .

For a non-negative random variable  $x$  (i.e.,  $p(x) = 0 \ \forall x < 0$ ) with mean  $\mu$ , the *Markov inequality* states that, for any  $\alpha > 0$ ,

$$P\{x \geq \alpha\} \leq \frac{\mu}{\alpha} \quad (1.5.4-17)$$

## 1.6 NOTES AND PROBLEMS

### 1.6.1 Bibliographical Notes

An excellent historical survey of the developments of estimation theory, starting from the least squares technique of Gauss and Legendre, the maximum likelihood technique of R. A. Fisher, the Wiener filtering for signals in noise, and the emergence of digital computer oriented recursive algorithms developed by Swerling, Kalman, Bucy, and others, can be found in [Sorenson85].

Other texts on the same topic as this one are [Meditch69, Sage71, Gelb74, Anderson79, Lewis86, Maybeck79, Maybeck82, Grewal93, Brown97].

For background material, further references on linear algebra as well as linear systems are [Zadeh63, Fortmann77, Chen84]. The existing literature on this topic consists of tens of books, and most of them are equally good. A succinct summary of this material can be found in [Gelb74]. The probabilistic tools are covered in [Papoulis84] and succinctly in [Gelb74]. Spectral factorization is discussed in [Poor88]. The statistical tools (hypothesis testing) can be found, for example, in [Melsa78, Johnston72], among others.

For the more specialized target tracking topics — in particular, tracking with uncertain origin observations and multitarget tracking — the reader is referred to [Blackman86, Bar-Shalom88]. The texts [Bar-Shalom95] and [Blackman99] are the most recent compilations of algorithms in this area, with suitable explanations and design guidelines.

### 1.6.2 Problems

**1-1 Independence versus conditional independence.** Consider  $x$  and  $y$  independent Gaussian random variables with zero mean and unity variance.

1. For an arbitrary new random variable does the following hold

$$p(x, y|z) = p(x|z)p(y|z) \quad ?$$

2. Let  $z = x + y$ . Write the explicit expression of the pdf  $p(x, y|z)$ . Use the Dirac delta function if necessary.

- 1-2 Monte Carlo runs for low probability events.** Given an experiment with binary outcome  $x$ , assume your colleague did a theoretical calculation that indicates that

$$P\{x = 1\} = \hat{p} = 10^{-4}$$

However, you do not believe this colleague and plan to conduct  $N$  Monte-Carlo (i.i.d.) repetitions of the experiment. The outcomes are  $x_i$ ,  $i = 1, \dots, N$ . Estimate, using a (say, 95%) confidence region, the number  $N$  of experiments you have to carry out to confirm or refute your colleague's theoretical calculation with, e.g., 10% tolerance.

- 1-3 Monte Carlo runs for algorithm comparison.** Two estimation algorithms are to be compared based on Monte Carlo runs. Assume their average performances (MSE) were obtained as  $\bar{C}^{(1)} = 5$  and  $\bar{C}^{(2)} = 8$  from  $N = 100$  runs.

1. Can one say that algorithm 1 is superior to algorithm 2?
2. Assume the standard deviations associated with these sample means are

$$\sigma_{\bar{C}^{(j)}} = \left[ \frac{1}{N^2} \sum_{i=1}^N [C_i^{(j)} - \bar{C}^{(j)}]^2 \right]^{1/2} = 1 \quad j = 1, 2$$

Using the above information answer question 1.

3. Denote the correlation coefficient between the outcomes of the runs with the two algorithms as  $\rho$ , that is,

$$E[(C_i^{(1)} - J^{(1)})(C_i^{(2)} - J^{(2)})] = \rho \sigma_{C^{(1)}} \sigma_{C^{(2)}}$$

Show that if  $\rho = 0.5$  and the variances are as in item 2, the optimal statistical test will yield a conclusive answer.

- 1-4 Covariance of a mixture.** Prove that an equivalent expression of the “spread of the means” term (1.4.16-9) in the covariance of a mixture is

$$\tilde{P} = \sum_i \sum_{j < i} p_i p_j (\bar{x}_i - \bar{x}_j)(\bar{x}_i - \bar{x}_j)'$$

where the summation is over all the pairs of indices without repetition.

#### 1-5 Dimension of a pdf.

1. Find the physical dimension of the pdf of the random vector  $x$  when  $x$  is
  - a. The position of a point in an  $n$ -dimensional Euclidean space.
  - b. The state of a constant (unknown) acceleration point moving along, say, the  $\xi$ -axis.
2. Find the units of the pdf of a velocity measured in furlongs/fortnight.
3. Can one compare the pdf of a random variable whose dimension is length to the joint pdf of two such random variables?

- 1-6 Log-normal random variable.** Given the random variable  $y \sim \mathcal{N}(\mu, \sigma^2)$ , then  $x \triangleq e^y$  is called a log-normal random variable. Find its mean and variance.

- 1-7 Moments of linear transformation of random variables.** Given the random variables  $x$  and  $y$  of dimensions  $n_x$  and  $n_y$ , with means  $\bar{x}$  and  $\bar{y}$ , respectively, and with covariances  $P_{xx}, P_{yy}, P_{xy}$ :

- Find the mean and covariance of the  $n_z$ -dimensional vector

$$z = Ax + By + c$$

where  $A$ , and  $B$  are matrices of appropriate dimensions.

- Indicate the dimensions of  $A$ ,  $B$ , and  $c$ .

- 1-8 Chapman-Kolmogorov equation.** Prove that the following equation holds for a (discrete time) Markov process

$$\int p[x(k)|x(k-1)]p[x(k-1)|x(k-2)] dx(k-1) = p[x(k)|x(k-2)]$$

- 1-9 Hypothesis testing with correlated noise.** Consider the hypotheses

$$H_0 : \theta = 0 \quad H_1 : \theta \neq 0$$

and the observations

$$z_i = \theta + w_i \quad i = 1, \dots, n$$

with  $w_i$  zero-mean jointly Gaussian but not independent. Denoting

$$w = [w_1 \ \cdots \ w_n]'$$

one has the covariance matrix (assumed given)

$$E[ww'] = P$$

For the above

- Specify the optimal hypothesis test for false alarm probability  $\alpha$ .
- Solve explicitly for  $n = 2$ ,  $P = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$  and  $\alpha = 1\%$ .

- 1-10 Partial derivative with respect to a matrix.** The partial derivative of a scalar  $q$  with respect to the matrix  $A = [a_{ij}]$  is defined as

$$\frac{\partial q}{\partial A} \triangleq \left[ \frac{\partial q}{\partial a_{ij}} \right]$$

Prove that

- For  $B$  symmetric,  $\frac{\partial}{\partial A} \text{tr}[ABA'] = 2AB$ .
- For  $B$  not symmetric,  $\frac{\partial}{\partial A} \text{tr}[AB] = B'$ .

- 1-11 Fourth joint moment of Gaussian random variables.** Show that, if the zero-mean scalar random variables  $x_i$ ,  $i = 1, \dots, 4$ , are jointly Gaussian, then

$$E[x_1x_2x_3x_4] = E[x_1x_2]E[x_3x_4] + E[x_1x_3]E[x_2x_4] + E[x_1x_4]E[x_2x_3]$$

*Hint:* Use the characteristic function (1.4.15-3) and take a suitable fourth derivative of it.

**1-12 Wiener process increments.**

- Using (1.4.20-3) show that

$$E[(dw(t))^2] = \alpha dt \quad \text{and} \quad dw(t) \sim \mathcal{N}(0, \alpha dt)$$

- Using the above, show that the limit of  $\frac{dw(t)}{dt}$  tends to  $\infty$ . Hint: Find the “order of magnitude” of  $dw(t)$  in terms of  $dt$  (a zero-mean random variable is of the order of its standard deviation).

**1-13 Conditional pdf of the sum of two Gaussian random variables.** Find the pdf (1.4.13-12) in the general case (1.4.13-6).**1-14 Probability matrix.** Find  $\sum_{j=1}^n \pi_{ij}$  for (1.4.22-9).**1-15 Laplacian pdf.** Given the random variable  $x$  with Laplacian (two-sided exponential) pdf

$$p(x) = c e^{-a|x|} \quad a > 0 \quad -\infty < x < \infty$$

- Find the normalization constant  $c$ , the mean and the variance of  $x$ .
- Repeat the above if it is known that  $|x| \leq b$ .

**1-16 Moments of a quadratic form with non-zero-mean random variables.** Consider the random variables  $x$  and  $y$  with means  $\bar{x}$  and  $\bar{y}$ , respectively, and with covariances  $P_{xx}, P_{yy}, P_{xy}$ . Evaluate  $E[x'Ax]$ .**1-17 The law of perversity of inanimate objects.** A buttered toast falls off a table. According to the “law of perversity of inanimate objects,” the probability of it falling on the floor with the buttered side down is greater than that with the buttered side up.

The motion (rotation and fall) of the toast can be modeled as follows.

Starting with the buttered side facing up, it acquires a rotational motion with a random angular rate  $\dot{\theta} \sim U[\dot{\theta}_1, \dot{\theta}_2]$ , assumed to stay constant until it hits the floor. The side which will hit the floor depends on the angle it rotated.

- Indicate the interval(s) of the rotation angle  $\theta$  for which the buttered side will hit the floor.
- Assuming a constant acceleration fall from the typical height of a table,  $h$ , with acceleration  $g = 10 \text{ m/s}^2$ , find the time the toast has to rotate.
- Find the pdf of the angle  $\theta$  it rotated until it hit the floor.
- Give the expression of the probability that the buttered side will be facing down when hitting the floor.
- Evaluate the above for  $h = 0.8 \text{ m}$ ,  $\dot{\theta}_1 = 200^\circ/\text{s}$ ,  $\dot{\theta}_2 = 300^\circ/\text{s}$ .
- Reevaluate the above for the land of the Brobdingnagians, in which the size of people (and their furniture) is scaled up by a factor of 9 compared to the above.

# **Chapter 2**

## **BASIC CONCEPTS IN ESTIMATION**

### **2.1 INTRODUCTION**

#### **2.1.1 Outline**

This chapter introduces some of the basic techniques of estimation that provide the foundation for state estimation and its applications like tracking, navigation, etc.

The problem of parameter estimation is defined in Section 2.2, where the two most commonly used models for unknown parameters (nonrandom and random) are also described. Section 2.3 deals with the maximum likelihood (ML) and the maximum a posteriori (MAP) estimators. The least squares (LS) and the minimum mean square error (MMSE) estimators are presented in Section 2.4.

The remaining sections deal with various “measures of quality” of estimators. Section 2.5 discusses unbiasedness and Section 2.6 discusses the variances of estimators. The consistency of estimators is discussed in Section 2.7, together with “information limit” results: the Cramer-Rao lower bound, the Fisher information, and estimator efficiency.

#### **2.1.2 Basic Concepts – Summary of Objectives**

Distinguish between

- Random parameters
- Nonrandom parameters

Define the following estimates

- Maximum likelihood
- Maximum a posteriori
- Least squares

- Minimum mean square error

Present “measures of quality” of estimators and “information limit” results

- Unbiasedness
- Variance
- Consistency
- The Cramer-Rao lower bound, Fisher information
- Efficiency

## 2.2 THE PROBLEM OF PARAMETER ESTIMATION

### 2.2.1 Definitions

The term **parameter** is used to designate a quantity (scalar or vector valued) that is assumed to be *time invariant*. If it does change with time, it can be designated (with a slight abuse of language) as a “time-varying parameter,” but its time variation must be “slow” compared to the state variables of a system. *State estimation*, which is for *dynamic systems*, is covered starting with Chapter 5.

The problem of estimating a (time invariant) parameter  $x$  is the following. Given the measurements

$$z(j) = h[j, x, w(j)] \quad j = 1, \dots, k \quad (2.2.1-1)$$

made in the presence of the disturbances (noises)  $w(j)$ , find a function of the  $k$  observations

$$\hat{x}(k) \triangleq \hat{x}[k, Z^k] \quad (2.2.1-2)$$

where these observations are denoted compactly as

$$Z^k \triangleq \{z(j)\}_{j=1}^k \quad (2.2.1-3)$$

that estimates the value of  $x$  in some sense.

The function (2.2.1-2) is called the **estimator**. The value of this function is the **estimate**. These terms, while not the same, will be used (sometimes) interchangeably.

The **estimation error** corresponding to the estimate  $\hat{x}$  is

$$\tilde{x} \triangleq x - \hat{x} \quad (2.2.1-4)$$

An alternate notation instead of (2.2.1-2) that will be used when  $k$  is fixed (and, therefore, can be omitted) is

$$\hat{x}(Z) \triangleq \hat{x}[k, Z^k] \quad (2.2.1-5)$$

where  $Z$  is the set of observations.

## Remark

Parameter estimation can be viewed as a ***static estimation problem***, while state estimation can be viewed as a ***dynamic estimation problem***.

### 2.2.2 Models for Estimation of a Parameter

There are two models one can use in the estimation of a (time invariant) parameter:

1. Nonrandom (“unknown constant”): There is an unknown true value  $x_0$ . This is also called the ***non-Bayesian*** or ***Fisher approach***.
2. Random: The parameter is a random variable with a *prior* (or *a priori*) pdf  $p(x)$  — a *realization* (see Subsection 1.4.2) of  $x$  according to  $p(x)$  is assumed to have occurred; this value then stays constant during the measurement process. This is also called the ***Bayesian approach***.

## The Bayesian Approach

In the ***Bayesian approach***, one starts with the *prior* pdf of the parameter from which one can obtain its *posterior* pdf (or a *posteriori* pdf) using ***Bayes' formula***:

$$p(x|Z) = \frac{p(Z|x)p(x)}{p(Z)} = \frac{1}{c}p(Z|x)p(x) \quad (2.2.2-1)$$

where  $c$  is the ***normalization constant***, which does not depend on  $x$ .

The posterior pdf can be used in several ways to estimate  $x$ .

## The Non-Bayesian (Likelihood Function) Approach

In contrast to the above, in the ***non-Bayesian approach*** there is no prior pdf associated with the parameter and thus one cannot define a posterior pdf for it.

In this case, one has the *pdf of the measurements conditioned on the parameter*, called the ***likelihood function (LF)*** of the parameter

$$\Lambda_Z(x) \triangleq p(Z|x) \quad (2.2.2-2)$$

or

$$\Lambda_k(x) \triangleq p(Z^k|x) \quad (2.2.2-3)$$

as a measure of how “likely” a parameter value is given the obtained observations. The likelihood function serves as a measure of the ***evidence from the data***.

The use of the LF (2.2.2-2) and a similar usage of the posterior pdf (2.2.2-1) are discussed in the next section.

## 2.3 MAXIMUM LIKELIHOOD AND MAXIMUM A POSTERIORI ESTIMATORS

### 2.3.1 Definitions of ML and MAP Estimators

#### Maximum Likelihood Estimator

A common method of estimating nonrandom parameters is the **maximum likelihood method** that maximizes the likelihood function (2.2.2-2). This yields the **maximum likelihood estimator (MLE)**

$$\boxed{\hat{x}^{\text{ML}}(Z) = \arg \max_x \Lambda_Z(x) = \arg \max_x p(Z|x)} \quad (2.3.1-1)$$

Note that, while  $x$  is an unknown constant,  $\hat{x}^{\text{ML}}(Z)$ , being a function of the set of random observations  $Z$ , is a random variable.

The MLE is the solution of the **likelihood equation**

$$\frac{d\Lambda_Z(x)}{dx} = \frac{dp(Z|x)}{dx} = 0 \quad (2.3.1-2)$$

#### Maximum A Posteriori Estimator

The corresponding estimate for a random parameter is the **maximum a posteriori (MAP)** estimator, which follows from the maximization of the posterior pdf (2.2.2-1):

$$\boxed{\hat{x}^{\text{MAP}}(Z) = \arg \max_x p(x|Z) = \arg \max_x [p(Z|x)p(x)]} \quad (2.3.1-3)$$

The last equality above follows from the fact that, when using Bayes' formula (2.2.2-1), the normalization constant is irrelevant for the maximization.

The MAP estimate, which depends on the observations  $Z$ , and through them on the realization of  $x$  is, obviously, a random variable.

### 2.3.2 MLE vs. MAP Estimator with Gaussian Prior

Consider the single measurement

$$z = x + w \quad (2.3.2-1)$$

of the unknown parameter  $x$  in the presence of the additive measurement noise  $w$ , assumed to be a normally (Gaussian) distributed random variable with mean zero and variance  $\sigma^2$ , that is,

$$w \sim \mathcal{N}(0, \sigma^2) \quad (2.3.2-2)$$

First assume that  $x$  is an unknown constant (no prior information about it is available). The likelihood function of  $x$  (denoted here without a subscript, for simplicity) is

$$\Lambda(x) = p(z|x) = \mathcal{N}(z; x, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-x)^2}{2\sigma^2}} \quad (2.3.2-3)$$

Then

$$\hat{x}^{\text{ML}} = \arg \max_x \Lambda(x) = z \quad (2.3.2-4)$$

since the peak or **mode** of (2.3.2-3) occurs at  $x = z$ .

Next assume that the prior information about the parameter is that  $x$  is Gaussian with mean  $\bar{x}$  and variance  $\sigma_0^2$ , that is,

$$p(x) = \mathcal{N}(x; \bar{x}, \sigma_0^2) \quad (2.3.2-5)$$

It is also assumed that  $x$  is independent of  $w$ .

Then the posterior pdf of  $x$  conditioned on the observation  $z$  is

$$p(x|z) = \frac{p(z|x)p(x)}{p(z)} = \frac{1}{c} e^{-\frac{(z-x)^2}{2\sigma^2} - \frac{(x-\bar{x})^2}{2\sigma_0^2}} \quad (2.3.2-6)$$

where

$$c = 2\pi\sigma\sigma_0 p(z) \quad (2.3.2-7)$$

is the normalization constant independent of  $x$ . This normalization constant, which guarantees that the pdf integrates to unity, is given explicitly next, after rearranging the exponent in (2.3.2-6).

After rearranging the exponent in the above by completing the squares in  $x$ , it can be easily shown in a manner similar to the one used in Subsection 1.4.14 that the posterior pdf of  $x$  is

$$p(x|z) = \mathcal{N}[x; \xi(z), \sigma_1^2] = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{|x-\xi(z)|^2}{2\sigma_1^2}} \quad (2.3.2-8)$$

i.e., Gaussian, where

$$\xi(z) \triangleq \frac{\sigma^2}{\sigma_0^2 + \sigma^2} \bar{x} + \frac{\sigma_0^2}{\sigma_0^2 + \sigma^2} z = \bar{x} + \frac{\sigma_0^2}{\sigma_0^2 + \sigma^2} (z - \bar{x}) \quad (2.3.2-9)$$

and (the “parallel resistors formula”)

$$\sigma_1^2 \triangleq \frac{\sigma_0^2 \sigma^2}{\sigma_0^2 + \sigma^2} \quad (2.3.2-10)$$

The maximization of (2.3.2-8) with respect to  $x$  yields immediately

$$\hat{x}^{\text{MAP}} = \xi(z) \quad (2.3.2-11)$$

that is,  $\xi(z)$  given by (2.3.2-9) is the *maximum a posteriori estimator* for the random parameter  $x$  with the prior pdf (2.3.2-5).

Note that the MAP estimator (2.3.2-9) for this (purely Gaussian) problem is a weighted combination of

1.  $z$ , the MLE, which is the peak (or mode) of the likelihood function;

2.  $\bar{x}$ , which is the peak of the prior pdf of the parameter to be estimated.

Equation (2.3.2-9) can be rewritten as follows:

$$\begin{aligned}\hat{x}^{\text{MAP}} &= (\sigma_0^{-2} + \sigma^{-2})^{-1} \sigma_0^{-2} \bar{x} + (\sigma_0^{-2} + \sigma^{-2})^{-1} \sigma^{-2} z \\ &= (\sigma_0^{-2} + \sigma^{-2})^{-1} \left[ \frac{\bar{x}}{\sigma_0^2} + \frac{z}{\sigma^2} \right]\end{aligned}\quad (2.3.2-12)$$

which indicates that the weightings of the prior mean and the measurement are *inversely proportional to their variances*.

Similarly, (2.3.2-10) can be rewritten as follows:

$$\sigma_1^{-2} = \sigma_0^{-2} + \sigma^{-2} \quad (2.3.2-13)$$

which shows that the *inverse variances* (also called **information** — this will be discussed in more detail in Subsection 3.4.2) are *additive*. This additivity property of information holds in general when the information sources are *independent*. (See also problem 2-8.)

### 2.3.3 MAP Estimator with One-Sided Exponential Prior

Consider the same problem as before except that the prior pdf of  $x$  is a **one-sided exponential pdf**

$$p(x) = ae^{-ax} \quad x \geq 0 \quad (2.3.3-1)$$

This can model, for instance, the arrival time in a stochastic process where the number of arrivals is Poisson distributed.

The ML estimate is the same as before in (2.3.2-4), that is,

$$\hat{x}^{\text{ML}} = z \quad (2.3.3-2)$$

The posterior pdf of  $x$  is now

$$p(x|z) = c(z) e^{-\frac{(z-x)^2}{2\sigma^2} - ax} \quad x \geq 0 \quad (2.3.3-3)$$

Since the exponent is quadratic in  $x$ , the above posterior pdf is Gaussian but truncated due to the fact that  $x$  cannot be negative as modeled by the prior given in (2.3.3-1).

In view of the fact that it cannot be negative, the maximizing argument of (2.3.3-3) is given by

$$\hat{x}^{\text{MAP}} = \max(z - \sigma^2 a, 0) \quad (2.3.3-4)$$

Note that the MAP estimate (2.3.3-4) in this case will always be smaller than the MLE (2.3.3-2) as long as the latter is not negative because the prior (2.3.3-1) attaches higher probability to smaller values of  $x$ .

### 2.3.4 MAP Estimator with Diffuse Prior

While  $\hat{x}^{\text{ML}}$  is based on a non-Bayesian approach and  $\hat{x}^{\text{MAP}}$  is based on the Bayesian approach, the latter will coincide with the former for a certain prior pdf, called a *diffuse pdf*.

This can be seen by rewriting the denominator in Bayes' formula

$$p(x|Z) = \frac{p(Z|x)p(x)}{p(Z)} \quad (2.3.4-1)$$

with the total probability theorem

$$p(Z) = \int_{-\infty}^{\infty} p(Z|x)p(x) dx \quad (2.3.4-2)$$

and assuming a *diffuse uniform prior pdf* for the parameter

$$p(x) = \epsilon \quad \text{for } |x| < \frac{1}{\epsilon} \quad (2.3.4-3)$$

over a “sufficiently large” region of length  $1/\epsilon$  where  $\epsilon > 0$  but small. Using (2.3.4-3) in (2.3.4-2) yields

$$p(Z) = \epsilon \int_{-\infty}^{\infty} p(Z|x)dx = \epsilon g(Z) \quad (2.3.4-4)$$

where  $g$  does not depend on  $x$ .

Then, inserting (2.3.4-4) into Bayes' formula (2.3.4-1) yields

$$p(x|Z) = \frac{p(Z|x)\epsilon}{\epsilon g(Z)} = \frac{p(Z|x)}{g(Z)} = \frac{1}{c} p(Z|x) \quad (2.3.4-5)$$

since  $\epsilon \neq 0$ .

This diffuse pdf is also called *improper pdf* because as  $\epsilon \rightarrow 0$ , it does not integrate to unity as a *proper pdf* does. Another name for it is *noninformative pdf* because it carries no information about the parameter: uniform distribution over an infinite interval at the limit.

A diffuse prior causes the posterior pdf of  $x$  to be proportional to its likelihood function and, thus, the MAP estimate to coincide with the MLE.

### Bayesian vs. Non-Bayesian Philosophies

The non-Bayesian MLE is, in view of the above discussion, nothing but the Bayesian MAP estimate with complete prior ignorance, reflected by the diffuse prior (2.3.4-3). *This provides a philosophically unifying view of the Bayesian and non-Bayesian approaches to estimation.*<sup>1</sup>

<sup>1</sup>Thus one can say that statisticians can be divided into two categories: Bayesians and closet Bayesians. This is a particular case of the following general taxonomy theorem: People can be divided into two categories. (*Proof:* The categories are those who believe that people can be divided into two categories and those who do not.)

**Remark**

In spite of the fact that the diffuse prior (2.3.4-3) does not integrate to unity and has no moments (i.e., it is *not a proper pdf*), the posterior pdf of  $x$  will be, in general, proper.

**Example**

Consider the problem of Subsection 2.3.2 where the prior (2.3.2-5) is made diffuse by making  $\sigma_0 \rightarrow \infty$ . A Gaussian pdf with very large variance becomes flat and at the limit looks like a uniform pdf over the whole real line.

When  $\sigma_0 \rightarrow \infty$  it can be seen from (2.3.2-9) that

$$\lim_{\sigma_0 \rightarrow \infty} \xi(z) = z \quad (2.3.4-6)$$

that is,  $\hat{x}^{\text{MAP}}$  coincides with  $\hat{x}^{\text{ML}}$ . This occurs regardless of the value of  $\bar{x}$ , which becomes irrelevant when  $\sigma_0 \rightarrow \infty$  (i.e., this is a *noninformative prior*). Thus the non-Bayesian approach can be seen as a degenerate case of the Bayesian approach.

**The Philosophical Meaning of the Prior**

The prior pdf assumed in a problem is in many cases the subjective assessment of phenomena. The uniform prior assumes Nature as “indifferent.” In game theory, Nature is assumed to be opposed to our interests. While neither of these two extreme points of view is correct, it is useful to keep in mind the well-known *principle of perversity of inanimate objects*.<sup>2</sup>

**2.3.5 The Sufficient Statistic and the Likelihood Equation**

If the likelihood function of a parameter can be decomposed as follows

$$\Lambda(x) \triangleq p(Z|x) = f_1[g(Z), x]f_2(Z) \quad (2.3.5-1)$$

then it is clear that the maximum likelihood estimate of  $x$  depends only on the function  $g(Z)$ , called the *sufficient statistic*, rather than on the entire data set  $Z$ .

The sufficient statistic *summarizes the information about  $x$  contained in the entire data*.

---

<sup>2</sup>As mentioned by Richard Bellman [Bellman61], this has been established by a number of experiments. The most conclusive of these involved dropping a piece of buttered toast on a rug. In 79.3% of the trials the toast fell buttered side down; for a mathematical proof of this principle, which does not hold in the land of the Brobdingnagians, see problem 1-17. For an extensive discussion on priors, see [Raiffa72].

**Example**

Consider the scalar measurements

$$z(j) = x + w(j) \quad j = 1, \dots, k \quad (2.3.5-2)$$

If the noise components  $w(j)$ ,  $j = 1, \dots, k$ , are independent and identically distributed zero-mean Gaussian random variables with variance  $\sigma^2$ , that is,

$$w(j) \sim \mathcal{N}(0, \sigma^2) \quad (2.3.5-3)$$

then

$$z(j) \sim \mathcal{N}(x, \sigma^2) \quad (2.3.5-4)$$

and, conditioned on  $x$ , the observations  $z(j)$  are mutually independent.

Thus, the **likelihood function** of  $x$  in terms of

$$Z^k \triangleq \{z(j), j = 1, \dots, k\} \quad (2.3.5-5)$$

is then

$$\begin{aligned} \Lambda_k(x) &\triangleq p(Z^k|x) \triangleq p[z(1), \dots, z(k)|x] \\ &= \prod_{j=1}^k \mathcal{N}[z(j); x, \sigma^2] = ce^{-\frac{1}{2\sigma^2} \sum_{j=1}^k |z(j)-x|^2} \end{aligned} \quad (2.3.5-6)$$

The likelihood function (2.3.5-6) can be rewritten into the product of two functions as in (2.3.5-1) as follows:

$$\begin{aligned} \Lambda_k(x) &= c e^{-\frac{1}{2\sigma^2} \sum_{j=1}^k z(j)^2 + \frac{1}{2\sigma^2} 2 \sum_{j=1}^k z(j)x - \frac{1}{2\sigma^2} kx^2} \\ &= c e^{-\frac{1}{2\sigma^2} \sum_{j=1}^k z(j)^2} e^{-\frac{1}{2\sigma^2} kx[x - \frac{2}{k} \sum_{j=1}^k z(j)]} \\ &\triangleq f_2(Z) f_1[g(Z), x] \end{aligned} \quad (2.3.5-7)$$

where

$$f_2(Z) \triangleq ce^{-\frac{1}{2\sigma^2} \sum_{j=1}^k z(j)^2} \quad (2.3.5-8)$$

$$f_1[g(Z), x] \triangleq e^{-\frac{1}{2\sigma^2} kx[x - 2\bar{z}]} \quad (2.3.5-9)$$

$$g(Z) \triangleq \frac{1}{k} \sum_{j=1}^k z(j) \triangleq \bar{z} \quad (2.3.5-10)$$

Thus, according to the definition (2.3.5-1),  $\bar{z}$  is the *sufficient statistic* for estimating  $x$ .

## The Likelihood Equation

To maximize the likelihood function (2.3.5-7), one sets its derivative with respect to  $x$  to zero. Since  $f_2$  is independent of  $x$ , the **likelihood equation** is

$$\frac{d\Lambda_k(x)}{dx} = 0 \iff \frac{df_1[g(Z), x]}{dx} = 0 \quad (2.3.5-11)$$

Since the logarithm is a monotonic transformation, equivalently one can use the derivative of the **log-likelihood function**

$$\frac{d \ln \Lambda_k(x)}{dx} = \frac{d \ln f_1[g(Z), x]}{dx} = -\frac{k}{2\sigma^2} 2(x - \bar{z}) = 0 \quad (2.3.5-12)$$

which yields

$$\hat{x}^{\text{ML}} = \bar{z} \quad (2.3.5-13)$$

The concept of sufficient statistic carries over to the MAP procedure in a completely analogous manner.

## 2.4 LEAST SQUARES AND MINIMUM MEAN SQUARE ERROR ESTIMATION

### 2.4.1 Definitions of LS and MMSE Estimators

#### The LS Estimator

Another common estimation procedure for nonrandom parameters is the **least squares (LS) method**. Given the (scalar and nonlinear) measurements

$$z(j) = h(j, x) + w(j) \quad j = 1, \dots, k \quad (2.4.1-1)$$

the **least squares estimator (LSE)** of  $x$  is, with notation (2.2.1-2),

$$\hat{x}^{\text{LS}}(k) = \arg \min_{\hat{x}} \left\{ \sum_{j=1}^k [z(j) - h(j, x)]^2 \right\}$$

(2.4.1-2)

This is the **nonlinear LS problem** — if the function  $h$  is linear in  $x$ , then one has the **linear LS problem**. The linear LS problem is considered in more detail for the vector case in Section 3.4.

The criterion in (2.4.1-2) makes no assumptions about the “measurement errors” or “noises”  $w(j)$ . If these are independent and identically distributed zero-mean Gaussian random variables, that is,

$$w(j) \sim \mathcal{N}(0, \sigma^2) \quad (2.4.1-3)$$

then the LSE (2.4.1-2) coincides with the MLE under these assumptions. In this case,

$$z(j) \sim \mathcal{N}[h(j, x), \sigma^2] \quad j = 1, \dots, k \quad (2.4.1-4)$$

The likelihood function of  $x$  is then

$$\begin{aligned} \Lambda_k(x) &\triangleq p(Z^k|x) \triangleq p[z(1), \dots, z(k)|x] \\ &= \prod_{j=1}^k \mathcal{N}[z(j); h(j, x), \sigma^2] = ce^{-\frac{1}{2\sigma^2} \sum_{j=1}^k [z(j) - h(j, x)]^2} \end{aligned} \quad (2.4.1-5)$$

and the minimization (2.4.1-2) is equivalent to the maximization of (2.4.1-5); that is, *the LS method is a “disguised” ML approach.*

## The MMSE Estimator

For random parameters, the counterpart of the above is the **minimum mean square error (MMSE) estimator**

$$\hat{x}^{\text{MMSE}}(Z) = \arg \min_{\hat{x}} E[(\hat{x} - x)^2|Z] \quad (2.4.1-6)$$

The solution to (2.4.1-6) is the **conditional mean** of  $x$

$$\boxed{\hat{x}^{\text{MMSE}}(Z) = E[x|Z] \triangleq \int_{-\infty}^{\infty} xp(x|Z) dx} \quad (2.4.1-7)$$

where the expectation is with respect to the conditional pdf (2.2.2-1).

The above follows by setting the derivative of (2.4.1-6) with respect to  $\hat{x}$  to zero:

$$\frac{d}{d\hat{x}} E[(\hat{x} - x)^2|Z] = E[2(\hat{x} - x)|Z] = 2(\hat{x} - E[x|Z]) = 0 \quad (2.4.1-8)$$

For vector random variables, (2.4.1-7) is obtained similarly by setting the gradient of the mean of the squared norm of the error to zero, that is

$$\nabla_{\hat{x}} E[(\hat{x} - x)'(\hat{x} - x)|Z] = 2(\hat{x} - E[x|Z]) = 0 \quad (2.4.1-9)$$

from which (2.4.1-7) follows immediately.

## Remarks

- With  $x$  being an unknown constant (nonrandom) and the noises in (2.4.1-1) modeled as random (not necessarily Gaussian), the LSE is a random variable.
- The MMSE estimate (2.4.1-7) is a random variable that depends on the observations  $Z$  and, through them, on (the realization of)  $x$ . Also, for a given  $Z$ ,  $x$  is a random variable with a conditional pdf (2.2.2-1).
- The MMSE estimation problem (2.4.1-6) is a particular case of Bayesian estimation where the expected value of a (positive definite) **cost function**  $C(\hat{x} - x)$  is to be minimized. The MMSE cost function is a quadratic. The widespread use of the quadratic criterion is due primarily to the (relative) ease of obtaining the solution. (See also problem 2-3.)

## 2.4.2 Some LS Estimators

### LS Estimator from a Single Measurement

For the problem of a single measurement of the unknown parameter  $x$ ,

$$z = x + w \quad (2.4.2-1)$$

the least squares criterion leads to

$$\hat{x}^{\text{LS}} = \arg \min_x [(z - x)^2] = z \quad (2.4.2-2)$$

which is the same result as  $\hat{x}^{\text{ML}}$  if  $w$  is zero-mean Gaussian. This is due to the fact that maximizing the likelihood function, which is a Gaussian pdf, is equivalent to minimizing the square in its exponent.

### LS Estimator from Several Measurements

Assume now that  $k$  measurements are made

$$z(j) = x + w(j) \quad j = 1, \dots, k \quad (2.4.2-3)$$

where  $w(j)$  are independent, identically distributed, normal, zero mean, and with common variance  $\sigma^2$ .

The likelihood function is, as in (2.4.1-5),

$$\Lambda_k(x) = ce^{-\frac{1}{2\sigma^2} \sum_{j=1}^k [z(j) - x]^2} \quad (2.4.2-4)$$

As before, the ML and LS estimates coincide and it can be easily shown that they are given by the following expression:

$$\hat{x}^{\text{ML}}(k) = \hat{x}^{\text{LS}}(k) = \frac{1}{k} \sum_{j=1}^k z(j) = \bar{z} \quad (2.4.2-5)$$

This estimate is known as the **sample mean** or **sample average**, since it estimates the unknown mean  $x$  of the  $k$  random variables from (2.4.2-3).

## 2.4.3 MMSE vs. MAP Estimator in Gaussian Noise

In the single measurement example with a prior pdf on the parameter to be estimated, discussed in Subsection 2.3.2, the posterior pdf of  $x$  was obtained in (2.3.2-8) as

$$p(x|z) = \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{|x-\xi(z)|^2}{2\sigma_1^2}} \quad (2.4.3-1)$$

It is apparent by inspection that the mean of this Gaussian pdf is  $\xi(z)$ , which is also the **mode** (peak) of this pdf.

Thus

$$\hat{x}^{\text{MMSE}} = E[x|z] = \xi(z) = \hat{x}^{\text{MAP}} \quad (2.4.3-2)$$

i.e., the MMSE estimator (the conditional mean) *coincides* with the MAP estimator.

This is due to the fact that the *mean* and the *mode* of a Gaussian pdf, which is symmetric and *unimodal*, coincide.

Note that, in view of (2.4.3-2), equation (2.4.3-1) can be also written as

$$p(x|z) = \mathcal{N}(x; \hat{x}^{\text{MMSE}}, \sigma_1^2) = \mathcal{N}(x; \hat{x}^{\text{MAP}}, \sigma_1^2) \quad (2.4.3-3)$$

## 2.5 UNBIASED ESTIMATORS

### 2.5.1 Definition

#### Non-Bayesian Case

For a nonrandom parameter, an estimator is said to be *unbiased* if

$$E[\hat{x}(k, Z^k)] = x_0 \quad (2.5.1-1)$$

where  $x_0$  is the true value of the parameter. The expectation in (2.5.1-1) is over the estimate, which is a random variable since it is a function of the measurements (2.2.1-3), and is taken with respect to the conditional pdf  $p(Z^k|x = x_0)$ .

#### Bayesian Case

If  $x$  is a random variable with a prior pdf  $p(x)$ , then the unbiasedness property is written as

$$E[\hat{x}(k, Z^k)] = E[x] \quad (2.5.1-2)$$

where the expectation on the left-hand side above is with respect to the joint pdf  $p(Z^k, x)$  and the one on the right-hand side is with respect to  $p(x)$ .

#### General Definition

The above unbiasedness requirements can be unified by requiring that the *estimation error*

$$\tilde{x} \triangleq x - \hat{x} \quad (2.5.1-3)$$

be zero mean, that is,

$$E[\tilde{x}] = 0 \quad (2.5.1-4)$$

Equation (2.5.1-4) covers both cases, with the expectation being taken over  $Z^k$  in the first case and over  $Z^k$  and  $x$  in the second case.

An estimator is unbiased if (2.5.1-4) holds for all  $k$  and is *asymptotically unbiased* if it holds in the limit as  $k \rightarrow \infty$ .

### 2.5.2 Unbiasedness of an ML and a MAP Estimator

Consider the ML estimator (2.3.2-4) of the parameter  $x$  with true value  $x_0$

$$\hat{x}^{\text{ML}} = z \quad (2.5.2-1)$$

from the single measurement

$$z = x + w \quad (2.5.2-2)$$

Its mean is

$$E[\hat{x}^{\text{ML}}] = E[z] = E[x_0 + w] = x_0 + E[w] = x_0 \quad (2.5.2-3)$$

since the mean of the Gaussian random variable  $w$  is zero.

For the MAP estimate (2.3.2-11) of  $x$  modeled as a Gaussian random variable with prior mean  $\bar{x}$ , prior variance  $\sigma^2$ , and independent of  $w$ ,

$$\hat{x}^{\text{MAP}} = \xi(z) \triangleq \frac{\sigma^2}{\sigma_0^2 + \sigma^2} \bar{x} + \frac{\sigma_0^2}{\sigma_0^2 + \sigma^2} z \quad (2.5.2-4)$$

one has

$$\begin{aligned} E[\hat{x}^{\text{MAP}}] &= E[\xi(z)] = \frac{\sigma^2}{\sigma_0^2 + \sigma^2} \bar{x} + \frac{\sigma_0^2}{\sigma_0^2 + \sigma^2} E[z] \\ &= \frac{\sigma^2}{\sigma_0^2 + \sigma^2} \bar{x} + \frac{\sigma_0^2}{\sigma_0^2 + \sigma^2} [\bar{x} + E(w)] = \bar{x} = E[x] \end{aligned} \quad (2.5.2-5)$$

Thus, both of these estimates are unbiased.

### 2.5.3 Bias in the ML Estimation of Two Parameters

Consider the problem of estimating the unknown mean  $x$  of a set of  $k$  measurements as in (2.4.2-3), with the additional parameter to be estimated being the variance  $\sigma^2$ , now also assumed to be unknown.

The likelihood function for the unknown parameters  $x$  and  $\sigma$  is

$$\Lambda_k(x, \sigma) = p[z(1), \dots, z(k)|x, \sigma] = \frac{1}{(2\pi)^{k/2}\sigma^k} e^{-\frac{1}{2\sigma^2} \sum_{j=1}^k [z(j)-x]^2} \quad (2.5.3-1)$$

To maximize the above, one writes the **likelihood equation** by setting to zero the derivatives of  $\Lambda$  or, more conveniently, of  $\ln \Lambda$ , with respect to  $x$  and  $\sigma$

$$\frac{\partial \ln \Lambda_k}{\partial x} = \frac{1}{\sigma^2} \sum_{j=1}^k [z(j) - x] = 0 \quad (2.5.3-2)$$

$$\frac{\partial \ln \Lambda_k}{\partial \sigma} = -\frac{k}{\sigma} + \frac{1}{\sigma^3} \sum_{j=1}^k [z(j) - x]^2 = 0 \quad (2.5.3-3)$$

## The Solution of the Likelihood Equation

The first equation yields  $\hat{x}^{\text{ML}}$  as before in (2.4.2-5), that is, the sample mean — in this problem the estimate of  $x$  is not affected at all by the fact that  $\sigma$  is also unknown. Substituting this into the second equation yields

$$\frac{\partial \ln \Lambda_k}{\partial \sigma} = -\frac{k}{\sigma} + \frac{1}{\sigma^3} \sum_{j=1}^k [z(j) - \hat{x}^{\text{ML}}]^2 = 0 \quad (2.5.3-4)$$

The resulting estimate, known as the **sample variance** based on  $k$  observations, is

$$[\hat{\sigma}^{\text{ML}}(k)]^2 = \frac{1}{k} \sum_{j=1}^k [z(j) - \hat{x}^{\text{ML}}]^2 = \frac{1}{k} \sum_{i=1}^k \left[ z(i) - \frac{1}{k} \sum_{i=1}^k z(i) \right]^2 \quad (2.5.3-5)$$

## The Means of the Sample Mean and Sample Variance

Denote the true values of the parameters by  $x_0$  and  $\sigma_0$ . The expected value of the sample mean is

$$E[\hat{x}^{\text{ML}}(k)] = E\left[\frac{1}{k} \sum_{j=1}^k z(j)\right] = x_0 \quad (2.5.3-6)$$

that is, the sample mean estimator (2.4.2-5) is unbiased.

The expected value of the sample variance (2.5.3-5) is

$$\begin{aligned} E\{[\hat{\sigma}^{\text{ML}}(k)]^2\} &= E\left\{\frac{1}{k} \sum_{j=1}^k \left[ z(j) - \frac{1}{k} \sum_{i=1}^k z(i) \right]^2\right\} \\ &= \frac{1}{k} \sum_{j=1}^k E\left\{ \left[ w(j) - \frac{1}{k} \sum_{i=1}^k w(i) \right]^2 \right\} \\ &= \frac{1}{k^3} \sum_{j=1}^k E\left\{ \left[ (k-1)w(j) - \sum_{\substack{i=1 \\ i \neq j}}^k w(i) \right]^2 \right\} \\ &= \frac{1}{k^2} [(k-1)^2 + k-1] \sigma_0^2 \\ &= \frac{k-1}{k} \sigma_0^2 \end{aligned} \quad (2.5.3-7)$$

Thus the sample variance (2.5.3-5) is *biased*, even though it becomes unbiased as  $k \rightarrow \infty$ , i.e., it is *asymptotically unbiased*. In order to be unbiased, the denominator in (2.5.3-5) should be  $k-1$  rather than  $k$ :

$$[\hat{\sigma}(k)]^2 = \frac{1}{k-1} \sum_{j=1}^k \left[ z(j) - \frac{1}{k} \sum_{i=1}^k z(i) \right]^2 \quad (2.5.3-8)$$

Expression (2.5.3-8) is the more common sample variance used. However, for reasonably large  $k$  this is not going to make a significant difference. (See also problem 2-4.)

## 2.6 THE VARIANCE AND MSE OF AN ESTIMATOR

### 2.6.1 Definitions of Estimator Variances

#### Non-Bayesian Case

For a non-Bayesian estimator,  $\hat{x}(Z)$ , (LS or ML) the **variance of the estimator** is

$$\text{var}[\hat{x}(Z)] \triangleq E[(\hat{x}(Z) - E[\hat{x}(Z)])^2] \quad (2.6.1-1)$$

where the averaging is over the observation set  $Z$ .

If this estimator is *unbiased*, that is,

$$E[\hat{x}(Z)] = x_0 \quad (2.6.1-2)$$

where  $x_0$  is the true value, then

$$\text{var}[\hat{x}(Z)] = E[(\hat{x}(Z) - x_0)^2] \quad (2.6.1-3)$$

If this estimator is *biased*, then (2.6.1-3) is its **mean square error (MSE)**<sup>3</sup>

$$\text{MSE}[\hat{x}(Z)] = E[(\hat{x}(Z) - x_0)^2] \quad (2.6.1-4)$$

#### Bayesian Case

For a Bayesian estimator, the **unconditional MSE** is<sup>4</sup>

$$\text{MSE}[\hat{x}(Z)] \triangleq E[(\hat{x}(Z) - x)^2] \quad (2.6.1-5)$$

where the averaging is with respect to the joint pdf of the observations  $Z$  and the random parameter  $x$ . The above can be rewritten, using the smoothing property of expectations (see Subsection 1.4.12), as follows:

$$\text{MSE}[\hat{x}(Z)] = E[E\{(\hat{x}(Z) - x)^2 | Z\}] = E[\text{MSE}[\hat{x}(Z)|Z]] \quad (2.6.1-6)$$

where the last expression inside the braces is the **conditional MSE**, i.e., for a given realization (or value) of the observations  $Z$ .

For the MMSE estimator, the conditional MSE is

$$\begin{aligned} E[(\hat{x}^{\text{MMSE}}(Z) - x)^2 | Z] &= E[(x - E(x|Z))^2 | Z] \\ &= \text{var}(x|Z) \end{aligned} \quad (2.6.1-7)$$

that is, the **conditional variance** of  $x$  given  $Z$ . Note that the expectations in (2.6.1-7) are with respect to  $p(x|Z)$ .

Averaging over  $Z$  yields

$$E[\text{var}(x|Z)] = E[(x - E(x|Z))^2] \quad (2.6.1-8)$$

which is the unconditional MSE (2.6.1-5) of the estimate  $\hat{x}^{\text{MMSE}}$ . This is the “average squared error over all the possible observations.”

<sup>3</sup>Mean square is a personality type from the official list compiled by psycho-statisticians.

<sup>4</sup>The variance, since it would be about the mean  $E[x]$  (if unbiased), has no real meaning.

## General Definition

With the definition of the estimation error

$$\tilde{x} \triangleq x - \hat{x} \quad (2.6.1-9)$$

one can say in a unified manner that the expected value of the square of the estimation error is the estimator's variance or MSE:

$$E[\tilde{x}^2] = \begin{cases} \text{var}(\hat{x}) & \text{if } \hat{x} \text{ is unbiased and } x \text{ is nonrandom} \\ \text{MSE}(\hat{x}) & \text{in all cases} \end{cases} \quad (2.6.1-10)$$

where the expectations are to be taken according to the discussion above.

The square root of the variance (or MSE) of an estimator

$$\sigma_{\hat{x}} \triangleq \sqrt{\text{var}(\hat{x})} \quad (2.6.1-11)$$

is its *standard error*, also called the *standard deviation associated with the estimator* or the *standard deviation of the estimation error*.

The standard error provides a measure of the accuracy of the estimator: assuming the estimation error to be Gaussian, the difference between the estimate and the true value will be up to 2 standard errors with 95% probability.

## 2.6.2 Comparison of Variances of an ML and a MAP Estimator

The “qualities” of the ML and MAP estimators discussed in Subsection 2.3.2 (from a single observation), as measured by their variances, will be compared next.

For the MLE given by (2.3.2-4) one has

$$\text{var}(\hat{x}^{\text{ML}}) = E[(\hat{x}^{\text{ML}} - x_0)^2] = E[(z - x_0)^2] \triangleq \sigma^2 \quad (2.6.2-1)$$

For the MAP estimate given by (2.3.2-11), which has a Gaussian prior in this case, one has<sup>5</sup>

$$\begin{aligned} \text{var}(\hat{x}^{\text{MAP}}) &= E[(\hat{x}^{\text{MAP}} - x)^2] \\ &= E\left\{\left[\frac{\sigma^2}{\sigma_0^2 + \sigma^2}\bar{x} + \frac{\sigma_0^2}{\sigma_0^2 + \sigma^2}(x + w) - x\right]^2\right\} \\ &= E\left[\left[\frac{\sigma^2}{\sigma_0^2 + \sigma^2}(\bar{x} - x) + \frac{\sigma_0^2}{\sigma_0^2 + \sigma^2}w\right]^2\right] \\ &= \frac{\sigma_0^2 \sigma^2}{\sigma_0^2 + \sigma^2} < \sigma^2 = \text{var}(\hat{x}^{\text{ML}}) \end{aligned} \quad (2.6.2-2)$$

---

<sup>5</sup>With abuse of notation because this is really the MSE; such abuses of notation are very common in the literature.

Thus it can be seen that the variance of the MAP estimator (given by the “parallel resistors formula”) is *smaller* than that of the MLE — this is due to the availability of *prior information*.

Note that in (2.6.2-1) the averaging is only over  $z$  (or, equivalently,  $w$ ) while in (2.6.2-2) the averaging is over  $w$  and  $x$ , which is assumed random. (See also problem 2-1.)

### 2.6.3 The Variances of the Sample Mean and Sample Variance

The **variance of the sample mean** (2.4.2-5) — the square of its **standard error** — is obtained as

$$E[(\hat{x}^{\text{ML}}(k) - x_0)^2] = E\left\{\left[\frac{1}{k} \sum_{j=1}^k [z(j) - x_0]\right]^2\right\} = \frac{\sigma^2}{k} \quad (2.6.3-1)$$

which, as  $k \rightarrow \infty$ , converges to zero; that is, this estimator is consistent (the definition is given in the next subsection).

The **variance of the sample variance** (2.5.3-5) is computed next. For simplicity, it is assumed that the mean is zero and known. The estimator of the variance in this case is

$$(\hat{\sigma}^{\text{ML}})^2 = \frac{1}{k} \sum_{j=1}^k z(j)^2 \quad (2.6.3-2)$$

and it can be easily shown that it is unbiased.

The variance of this estimator is, with the true value denoted by  $\sigma^2$ ,

$$\begin{aligned} E[((\hat{\sigma}^{\text{ML}})^2 - \sigma^2)^2] &= E\left\{\left[\frac{1}{k} \sum_{j=1}^k z(j)^2 - \sigma^2\right]^2\right\} \\ &= \frac{1}{k^2} \sum_{j=1}^k \sum_{i=1}^k E[w(j)^2 w(i)^2] - 2\sigma^2 \frac{1}{k} \sum_{j=1}^k E[w(j)^2] + \sigma^4 \\ &= \frac{1}{k^2} [k(k-1)\sigma^4 + k3\sigma^4] - \frac{2}{k} k\sigma^4 + \sigma^4 = \frac{2\sigma^4}{k} \end{aligned} \quad (2.6.3-3)$$

which also converges to zero as  $k \rightarrow \infty$ .

The following relationship has been used in (2.6.3-3)

$$E[w(i)^2 w(j)^2] = \begin{cases} \sigma^4 & \text{if } i \neq j \\ 3\sigma^4 & \text{if } i = j \end{cases} \quad (2.6.3-4)$$

The fourth moment of  $w$  is needed here and, assuming it to be Gaussian, use was made of (1.4.15-7).

The **standard error of the sample variance** from  $k$  samples is therefore, from (2.6.3-3), given by

$$\sigma_{(\hat{\sigma}^{\text{ML}})^2} = \sigma^2 \sqrt{2/k} \quad (2.6.3-5)$$

In the above the notation,  $\sigma_\xi$  has been used to denote the **standard error of the estimate**  $\xi$ .

## Application — The Number of Samples Needed to Estimate a Variance with a Given Accuracy

Based on this result, the number of samples needed to obtain the sample variance *within 10% of the true value with probability of 95%* can be obtained as follows.

Assuming for convenience that the sample variance is normally distributed<sup>6</sup> about its mean (equal to the true variance) with standard error as above, one has the 95% confidence region

$$P \left\{ |(\hat{\sigma}^{\text{ML}})^2 - \sigma^2| \leq 1.96\sigma^2\sqrt{2/k} \right\} = 0.95 \quad (2.6.3-6)$$

The requirement of at most 10% error in the variance estimate, that is,

$$\frac{|(\hat{\sigma}^{\text{ML}})^2 - \sigma^2|}{\sigma^2} = 0.1 \quad (2.6.3-7)$$

leads to setting

$$1.96\sqrt{2/k} = 0.1 \quad (2.6.3-8)$$

which yields

$$k \approx 800 \quad (2.6.3-9)$$

The resulting very large number of samples necessary for the required accuracy justifies the use of the CLT in (2.6.3-6). (See also problem 1-2.)

## 2.6.4 Estimation of the Probability of an Event

The **estimation of the probability of an event** can be done as follows. If in  $N$  independent identical experiments an event occurs  $N_0$  times, then one can use the following estimate of the probability  $p$  of this event:

$$\hat{p} = \frac{N_0}{N} \quad (2.6.4-1)$$

It can be easily seen that the above is an unbiased estimate since

$$E\hat{p} = E \frac{N_0}{N} = \frac{Np}{N} = p \quad (2.6.4-2)$$

Now, since  $N_0$  is a Bernoulli random variable (the sum of  $N$  i.i.d. binary random variables), its distribution is given exactly by the binomial distribution. However, one can gain more insight into this by assuming  $N$  large enough, in which case one can use, in view of the CLT, the Gaussian approximation of the pdf of  $\hat{p}$ . ICBES that

$$\text{var}[\hat{p}] = \frac{p(1-p)}{N} \quad (2.6.4-3)$$

---

<sup>6</sup>In view of the CLT, see Subsection 1.4.23.

Thus, using the 95% ( $2\sigma$ ) confidence region, one can say that

$$P \left\{ |\hat{p} - p| \leq 2 \sqrt{\frac{p(1-p)}{N}} \right\} = 0.95 \quad (2.6.4-4)$$

Since we do not know  $p$ , one can use  $\hat{p}$  instead of  $p$  in the expression of the variance above.

As an example, if  $N = 1000$  and  $\hat{p} = 0.1$ , the standard deviation of the estimate is  $\sqrt{0.1 \cdot 0.9/1000} \approx 0.01$ . Therefore, the above confidence region becomes  $[0.08, 0.12]$ .

Such techniques are commonly used in public opinion polls, but in a more conservative way: The worst case of  $p = 0.5$  is assumed for the variance. For  $N = 1000$  the standard deviation of the estimate is then  $\sqrt{0.5 \cdot 0.5/1000} \approx 0.016$  with the confidence region being  $[\hat{p} - 0.03, \hat{p} + 0.03]$ . This is how to interpret the results from a poll, which state a **margin of error** of  $\pm 3\%$ .

## 2.7 CONSISTENCY AND EFFICIENCY OF ESTIMATORS

### 2.7.1 Consistency

An estimator of a *nonrandom parameter* is said to be a **consistent estimator** if the estimate (which is a random variable) converges to the true value in some stochastic sense.<sup>7</sup> Using the **convergence in mean square** criterion, then

$$\lim_{k \rightarrow \infty} E [(\hat{x}(k, Z^k) - x_0)^2] = 0 \quad (2.7.1-1)$$

is the condition for **consistency in the mean square sense**. The expectation is taken over  $Z^k$ , as in (2.5.1-1).

For a *random parameter*, convergence of its estimator in the mean square sense requires

$$\lim_{k \rightarrow \infty} E [(\hat{x}(k, Z^k) - x)^2] = 0 \quad (2.7.1-2)$$

where the expectation is over  $Z^k$  and  $x$ , as in (2.5.1-2).

Similarly to the unbiasedness case, consistency can be expressed as the requirement that the estimation error converge to zero, that is,

$$\lim_{k \rightarrow \infty} \tilde{x}(k, Z^k) = 0 \quad (2.7.1-3)$$

in some stochastic (e.g., mean square) sense.

---

<sup>7</sup>Another definition of consistency is “the last refuge of the unimaginative” (Oscar Wilde).

## Remark

The consistency defined above is an *asymptotic* property; that is, it is defined for the case when the sample size  $k$  tends to infinity and the object of estimation is a *fixed parameter*. Later, in the context of state estimation, where the object of estimation is an *evolving state*, there will be another definition of consistency as a *finite sample size* property.

### 2.7.2 The Cramer-Rao Lower Bound and the Fisher Information Matrix

According to the **Cramer-Rao lower bound (CRLB)**, the mean square error corresponding to the estimator of a parameter *cannot be smaller* than a certain quantity related to the likelihood function.

#### Scalar Case

In the estimation of a scalar *nonrandom* parameter  $x$  with an *unbiased* estimator  $\hat{x}(Z)$ , the variance is bounded from below as follows:

$$\boxed{E [(\hat{x}(Z) - x_0)^2] \geq J^{-1}} \quad (2.7.2-1)$$

where

$$J \triangleq -E \left[ \frac{\partial^2 \ln \Lambda(x)}{\partial x^2} \right]_{x=x_0} = E \left\{ \left[ \frac{\partial \ln \Lambda(x)}{\partial x} \right]^2 \right\}_{x=x_0} \quad (2.7.2-2)$$

is the **Fisher information**,  $\Lambda(x) = p(Z|x)$  is the likelihood function (2.2.2-2) denoted for simplicity without subscript, and  $x_0$  is the true value of the unknown constant  $x$ .

For a scalar *random* parameter  $x$  estimated by an unbiased estimator  $\hat{x}(Z)$ , the variance is bounded from below by a similar expression, namely,

$$E [(\hat{x}(Z) - x)^2] \geq J^{-1} \quad (2.7.2-3)$$

where

$$J \triangleq -E \left[ \frac{\partial^2 \ln p(Z, x)}{\partial x^2} \right] = E \left\{ \left[ \frac{\partial \ln p(Z, x)}{\partial x} \right]^2 \right\} \quad (2.7.2-4)$$

The expectations in (2.7.2-2) and (2.7.2-4) are taken as in (2.7.1-1) and (2.7.1-2), respectively.

Note that the Fisher information has two forms in (2.7.2-2) as well as in (2.7.2-4). The proof of (2.7.2-1) and the equivalence of the two forms in (2.7.2-2) — one with first partial derivatives and the other with second partials — is given later.

If an estimator's variance is equal to the CRLB, then such an estimator is called **efficient**.

## Multidimensional Case

For *nonrandom vector parameters*, the CRLB states that the covariance matrix of an unbiased estimator is bounded from below as follows:

$$E[[\hat{x}(Z) - x_0][\hat{x}(Z) - x_0]'] \geq J^{-1} \quad (2.7.2-5)$$

where the *Fisher information matrix (FIM)* is

$$J \triangleq -E[\nabla_x \nabla_x' \ln \Lambda(x)]|_{x=x_0} = E[[\nabla_x \ln \Lambda(x)][\nabla_x \ln \Lambda(x)']]|_{x=x_0} \quad (2.7.2-6)$$

and  $x_0$  is the true value of the vector parameter  $x$ .

As in the scalar case, note the two forms of the FIM: one with the Hessian of the log-likelihood function and the other with the dyad of its gradient.

The matrix inequality in (2.7.2-5) is to be interpreted as follows:

$$A \geq B \iff C \triangleq A - B \geq 0 \quad (2.7.2-7)$$

that is, the difference  $C$  of the two matrices is positive semidefinite.

A similar expression holds for the case of a multidimensional random parameter.

## Remarks

The FIM can be seen as a quantification of the (maximum) *existing information* in the data about a parameter. Efficiency amounts to the *extracted information* being equal to the existing one, i.e., all the information has been extracted.

A necessary condition for an estimator to be consistent in the mean square sense is that there must be an increasing amount of information (in the sense of Fisher) about the parameter in the measurements — the Fisher information has to tend to infinity as  $k \rightarrow \infty$ . Then the CRLB converges to zero as  $k \rightarrow \infty$  and thus the variance can also converge to zero.

## Note

For estimators that are *biased*, there is a modified version of the CRLB (e.g., [Van Trees68]).

### 2.7.3 Proof of the Cramer-Rao Lower Bound

Let  $\hat{x}(z)$  be an unbiased estimate of the nonrandom real-valued parameter  $x$  based on the observation (or set of observations) denoted now as  $z$ . The likelihood function of  $x$  is

$$\Lambda(x) = p(z|x) \quad (2.7.3-1)$$

It will be assumed that the first and second derivatives of (2.7.3-1) with respect to  $x$  exist and are absolutely integrable.

From the unbiasedness condition on the estimate  $\hat{x}(z)$ , one has (the true value is denoted now also as  $x$ )

$$E[\hat{x}(z) - x] = \int_{-\infty}^{\infty} [\hat{x}(z) - x] p(z|x) dz = 0 \quad (2.7.3-2)$$

The derivative of the above with respect to  $x$  is

$$\begin{aligned} \frac{d}{dx} \int_{-\infty}^{\infty} [\hat{x}(z) - x] p(z|x) dz &= \int_{-\infty}^{\infty} \frac{\partial}{\partial x} \{ [\hat{x}(z) - x] p(z|x) \} dz \\ &= - \int_{-\infty}^{\infty} p(z|x) dz + \int_{-\infty}^{\infty} [\hat{x}(z) - x] \frac{\partial p(z|x)}{\partial x} dz \\ &= 0 \end{aligned} \quad (2.7.3-3)$$

Using the fact that the first integral in the last line above is equal to unity and the identity

$$\frac{\partial p(z|x)}{\partial x} = \frac{\partial \ln p(z|x)}{\partial x} p(z|x) \quad (2.7.3-4)$$

yields from (2.7.3-3)

$$\int_{-\infty}^{\infty} [\hat{x}(z) - x] \frac{\partial \ln p(z|x)}{\partial x} p(z|x) dz = 1 \quad (2.7.3-5)$$

Equation (2.7.3-5) can be rewritten as

$$\int_{-\infty}^{\infty} \left\{ [\hat{x}(z) - x] \sqrt{p(z|x)} \right\} \left\{ \frac{\partial \ln p(z|x)}{\partial x} \sqrt{p(z|x)} \right\} dz = 1 \quad (2.7.3-6)$$

The **Schwarz inequality** for real-valued functions, which is a generalized version of (1.3.4-2), is

$$|\langle f_1, f_2 \rangle| \triangleq \int_{-\infty}^{\infty} f_1(z) f_2(z) dz \leq \|f_1\| \|f_2\| \quad (2.7.3-7)$$

where

$$\|f_i\| \triangleq \{\langle f_i, f_i \rangle\}^{1/2} = \left\{ \int_{-\infty}^{\infty} f_i(z)^2 dz \right\}^{1/2} \quad (2.7.3-8)$$

The equality in (2.7.3-7) holds if and only if

$$f_1(z) = c f_2(z) \quad \forall z \quad (2.7.3-9)$$

Note that the left-hand side of (2.7.3-6) is an inner product of two functions as in (2.7.3-7). Using (2.7.3-7) to majorize the left-hand side of (2.7.3-6) yields

$$\left\{ \int_{-\infty}^{\infty} [\hat{x}(z) - x]^2 p(z|x) dz \right\}^{1/2} \left\{ \int_{-\infty}^{\infty} \left[ \frac{\partial \ln p(z|x)}{\partial x} \right]^2 p(z|x) dz \right\}^{1/2} \geq 1 \quad (2.7.3-10)$$

which can be rewritten as

$$E\{[\hat{x}(z) - x]^2\} \geq \left\{ E \left[ \frac{\partial \ln p(z|x)}{\partial x} \right]^2 \right\}^{-1} \quad (2.7.3-11)$$

with equality holding if and only if

$$\frac{\partial \ln p(z|x)}{\partial x} = c(x)[\hat{x}(z) - x] \quad \forall z \quad (2.7.3-12)$$

Equation (2.7.3-11) is equivalent to (2.7.2-1), which completes the proof of the CRLB for a nonrandom scalar parameter. In view of (2.7.3-2), which holds at the true value of the parameter, all the partial derivatives are to be evaluated at the true value of the parameter, which is indicated explicitly only in (2.7.2-2).

### Equivalence of the Two Forms of the Fisher Information

To prove the equivalence of the two forms of the Fisher information in (2.7.2-2), consider the identity

$$\int_{-\infty}^{\infty} p(z|x) dz = 1 \quad (2.7.3-13)$$

Taking the derivative of the above with respect to  $x$  yields

$$\int_{-\infty}^{\infty} \frac{\partial p(z|x)}{\partial x} dz = 0 \quad (2.7.3-14)$$

Using identity (2.7.3-4), the above can be rewritten as

$$\int_{-\infty}^{\infty} \frac{\partial \ln p(z|x)}{\partial x} p(z|x) dz = 0 \quad (2.7.3-15)$$

Taking now the derivative of (2.7.3-15) with respect to  $x$  leads to

$$\int_{-\infty}^{\infty} \frac{\partial^2 \ln p(z|x)}{\partial x^2} p(z|x) dz + \int_{-\infty}^{\infty} \left[ \frac{\partial \ln p(z|x)}{\partial x} \right]^2 p(z|x) dz = 0 \quad (2.7.3-16)$$

which proves the equivalence of the expression of the Fisher information with the second partial derivative of the log-likelihood function with the one that has the square of the first partial derivative, as in (2.7.2-2).

### 2.7.4 An Example of Efficient Estimator

Consider the likelihood function (2.4.2-4) for the estimation of the mean  $x$  from a set of  $k$  independent and identically distributed measurements with Gaussian noises.

The Fisher information in this case is the scalar quantity

$$J = -E \left[ \frac{\partial^2 \ln \Lambda_k(x)}{\partial x^2} \right]_{x=x_0} = \frac{k}{\sigma^2} \quad (2.7.4-1)$$

Thus

$$E [(\hat{x}^{\text{ML}}(k) - x_0)^2] \geq J^{-1} = \frac{\sigma^2}{k} \quad (2.7.4-2)$$

Comparing the above to (2.6.3-1), it is seen that the CRLB is met; that is, the ML estimator (which, in this case, is the sample mean) is efficient. This is because condition (2.7.3-12) is satisfied.

Since the variance (2.7.4-2) converges to zero as  $k \rightarrow \infty$ , this estimator is also consistent. (See also problem 2-2.)

### Evaluation of the CRLB

In the simple case considered above, the expression of  $J$  is independent of  $x$ . In general this is not true and there is need to evaluate  $J$  at the *true value of  $x$* .

If the true value of the parameter is not available, then the *evaluation of the CRLB*, which amounts to a linearization, is done *at the estimate*. Caution has to be exercised in this case, since the unavoidable estimation errors can lead to a *possibly incorrect value* of the resulting Fisher information  $J$ , which, in general, is a matrix.

### 2.7.5 Large Sample Properties of the ML Estimator

The following are the *large-sample properties of the ML estimator*:

1. It is *asymptotically unbiased*.
2. It is *asymptotically efficient*.

Thus, if there is “enough information” in the measurements, in which case the CRLB will tend to zero, the variance of the ML estimate will also converge to zero. Therefore, the ML estimate will converge to the true value — it will be consistent.

Another property of the ML estimator is the following:

3. It is *asymptotically Gaussian*.

Combining all the above, *the ML estimate is asymptotically Gaussian with the mean equal to the true value of the parameter to be estimated and variance given by the CRLB*.

This can be summarized for a vector parameter as

$$\hat{x}^{\text{ML}}(k) \sim \mathcal{N}(x, J^{-1}) \quad \text{for large } k \quad (2.7.5-1)$$

where  $J$  is the Fisher information matrix.

Comparing the (non-Bayesian) MLE (2.7.5-1) with the (Bayesian) MMSE estimate, the conditional mean

$$\hat{x}^{\text{MMSE}} = E[x|z] \quad (2.7.5-2)$$

points out the contrast between these two philosophies:

1. In (2.7.5-1), given  $x$ , the estimate  $\hat{x}^{\text{ML}}$  is a random variable, function of  $z$ ;
2. In (2.7.5-2), given  $z$ , the true value  $x$  is a random variable.

## 2.8 SUMMARY

### 2.8.1 Summary of Estimators

**Estimator of a parameter** — a function of the measurements that yields a “best approximation” for the value of a parameter.

**Estimate of a parameter** — the value taken by the estimator for the given values (realizations) of the measurements.

Models for the parameter to be estimated:

1. *Unknown constant (nonrandom).*
2. *Random:* a (single) realization of a random variable according to a certain prior pdf.

Model 2 yields the *Bayesian approach*, whereas model 1 leads to what is called the *non-Bayesian approach*.

**Likelihood function** of a (nonrandom) parameter — pdf of the measurements conditioned on the parameter.

**Bayes' formula** — given a prior pdf of a (random) parameter, this formula yields its posterior pdf conditioned on the measurements.

**ML estimate** (of a nonrandom parameter) — the value of the parameter that maximizes its likelihood function.

**MAP estimate** (of a random parameter) — the value of the parameter that maximizes its posterior pdf.

The MAP estimate of a parameter with a *diffuse (noninformative)* prior pdf coincides with its MLE.

**LS estimate** (of a nonrandom parameter) — minimizes the sum of the squares of the errors between the measurements and the observed function of the parameter.

**MMSE estimate** (of a random parameter) — minimizes the expected value (mean) of the square of the parameter estimation error conditioned on the measurements. This estimate is the *conditional mean* of the parameter given the measurements.

If in a given set of measurements the errors are additive, zero mean, *Gaussian*, and independent, then the *LS* estimate coincides with the *ML* estimate.

The *MAP* estimate of a *Gaussian* random variable coincides with its *MMSE* estimate (conditional mean).

## 2.8.2 Summary of Estimator Properties

**Unbiased estimator** — if the mean of the corresponding error is zero.

**Variance/MSE of an estimator** — the expected value of the square of the estimation error of an unbiased/biased estimator. The variance of the estimator of a parameter modeled as random (with some prior) is *smaller* than when it is modeled as an unknown constant.

**Consistent estimator** — if the corresponding error converges to zero in some stochastic sense (most common: in mean square).

**CRLB** — lower bound on the achievable variance in the estimation of a parameter. For an unbiased estimator, it is given by the inverse of the *Fisher information matrix (FIM)*.

**FIM** — quantifies the existing total information about the parameter of interest in the observations.

**Efficient estimator** — if its variance meets the CRLB, that is, if all the existing information has been extracted.

## On the Terminology

In (most of) the literature there is little or no distinction between the terms LS and MMSE estimation. The MMSE estimation is sometimes called LS, which is incorrect according to our definition, or *least mean square (LMS)*, which is a valid alternate designation. Another term used is *minimum variance (MV)*.

## 2.9 NOTES AND PROBLEMS

### 2.9.1 Bibliographical Notes

The basic concepts in estimation are discussed, for example, in [Van Trees68, Sage71, Melsa78]. The proof of the CRLB for vector-valued parameters can be found in [Van Trees68] and [Ljung87, p. 206].

Another model of uncertainty in parameter estimation is the “unknown but bounded” approach discussed in [Schweppe73].

### 2.9.2 Problems

- 2-1 Estimators for a discrete-valued parameter.** A discrete-valued parameter with the prior pdf

$$p(x) = \sum_{i=1}^2 p_i \delta(x - i)$$

is measured with the additive noise  $w \sim \mathcal{N}(0, \sigma^2)$

$$z = x + w$$

1. Find the posterior pdf of the parameter.
2. Find its MAP estimate and the associated MSE conditioned on  $z$ .
3. Find its MMSE estimate and the associated variance.
4. Evaluate these estimates and MSE for

Case	$p_1$	$\sigma$	$z$
A	0.5	1	1.5
B	0.5	1	3
C	0.3	1	1.5
D	0.5	0.1	1.8

5. Comment on the meaningfulness of the two estimates in the above four cases.

- 2-2 ML Estimation with correlated noises.** A parameter  $x$  is measured with correlated rather than independent additive Gaussian noises

$$z_k = x + w_k \quad k = 1, \dots, n$$

with

$$E[w_k] = 0 \quad E[w_k w_j] = \begin{cases} 1 & k = j \\ \rho & |k - j| = 1 \\ 0 & |k - j| > 1 \end{cases}$$

For  $n = 2$ :

1. Write the likelihood function of the parameter  $x$ .
2. Find the MLE of  $x$ . What happens if  $\rho = 1$ ? What happens if  $\rho = -1$ ?
3. Find the CRLB for the estimation of  $x$ . Show the effect of  $\rho > 0$  versus  $\rho < 0$ . Explain what happens at  $\rho = -1$ .
4. Is the MLE efficient? Can one have a perfect (zero-variance) estimate?

(The remaining items are more challenging.) For general  $n$ , let

$$z \triangleq [z_1 \dots z_n]' \quad \mathbf{1} \triangleq [1 \dots 1]' \quad w \triangleq [w_1 \dots w_n]' \quad P \triangleq E[ww']$$

5. Using the above notations, write the likelihood function of  $x$ .
6. Find the MLE of  $x$ .

**2-3 Estimation criteria that lead to the conditional mean.** Show that, in estimating a random vector  $x$  with the following criteria

1.  $\min_{\hat{x}} E[(x - \hat{x})' A(x - \hat{x})|z]$ ,  $\forall A > 0$  (positive definite)
2.  $\min_{\hat{x}} \text{tr}[P]$  with  $P \triangleq E[(x - \hat{x})(x - \hat{x})'|z]$
3.  $\min_{\hat{x}} \text{tr}[AP]$  with  $A$  and  $P$  as above

all yield the same result  $\hat{x} = E[x|z]$ .

**2-4 Estimate of the variance with the smallest MSE.** Consider the problem of estimating the mean and the variance of a set of independent and identically distributed Gaussian random variables  $z(j)$ ,  $j = 1, \dots, k$ , as in Subsection 2.5.3, with the true mean  $x_0$  and true variance  $\sigma_0^2$ .

1. Show that the value of  $n$  in

$$[\hat{\sigma}(k, n)]^2 = \frac{1}{n} \sum_{j=1}^k \left[ z(j) - \frac{1}{k} \sum_{i=1}^k z(i) \right]^2$$

that minimizes the MSE of the above (defined according to (2.6.1-4) with respect to  $\sigma_0^2$ ) is  $n = k + 1$ .

2. Can a biased estimator have a smaller MSE than an unbiased one?

**2-5 MAP estimate with two-sided exponential (Laplacian) prior pdf.** Consider the same problem as in Subsection 2.3.2 but with a two-sided exponential prior

$$p(x) = \frac{a}{2} e^{-a|x|}$$

1. Write the posterior pdf of  $x$ .
2. Find  $\hat{x}^{\text{MAP}}$ .

**2-6 Two-sided exponential prior made diffuse.**

1. Specify the limiting process that will make the prior from problem 2-5 into a diffuse one.
2. Show that the resulting MAP estimate coincides with the MLE.

**2-7 Minimum magnitude error estimate.** Given  $p(x, z)$ , show that the Bayesian estimation that minimizes the expected value of the cost function

$$C(x - \hat{x}) \triangleq |x - \hat{x}|$$

yields  $\hat{x} = x_m$ , the **median** of  $x$ , defined as

$$\int_{-\infty}^{x_m} p(x|z) dx = \frac{1}{2}$$

- 2-8 MAP with Gaussian prior — vector version.** Given  $z = x + w$ , where all the variables are  $n$ -vectors, with

$$w \sim \mathcal{N}(0, P) \quad x \sim \mathcal{N}(\bar{x}, P_0)$$

and  $x$  independent of  $w$ . Find the MAP estimator of  $x$  in terms of  $z$  and the covariance of this estimator.

- 2-9 Conditional variance versus unconditional variance.** Let

$$\begin{aligned}\bar{x} &\triangleq E[x] & \text{var}(x) &\triangleq E[(x - \bar{x})^2] \\ \hat{x} &\triangleq E[x|z] & \text{var}(x|z) &\triangleq E[(x - \hat{x})^2|z]\end{aligned}$$

Prove that

$$\text{var}(x) \geq E[\text{var}(x|z)]$$

- 2-10 MMSE with exponential prior.** Given the prior pdf of  $x$  as  $p(x) = e^{-x}$ ,  $x \geq 0$  and the observation  $z = x + w$  where  $w \sim \mathcal{N}(0, 1)$  and independent of  $x$ , find the following:

1.  $p(z|x)$
2.  $p(x|z)$
3.  $E[x|z]$
4.  $\text{var}[x|z]$

- 2-11 Altitude estimation from slant range.** A sensor is located at  $(0, 0)$ . It is desired to estimate the height (altitude)  $y$  of a point (an aircraft) located at  $(d, y)$ , where  $d$  (the horizontal range) is known, based on the “slant range” measurement  $z = r + w$  where  $r = h(y) = \sqrt{d^2 + y^2}$  and  $w \sim \mathcal{N}(0, \sigma^2)$ .

1. Write the likelihood function of  $y$ .
2. Find the CRLB for estimating  $y$ .
3. Evaluate the standard deviation of the estimate according to the CRLB for  $d = 10^5$ ,  $\sigma = 10^2$ , and assumed true value  $y = 10^3$ . How useful would such an estimate be?
4. Find the expression of the MLE of  $y$  in terms of  $z$  and  $d$ .

- 2-12 Superefficiency?** A scalar parameter is estimated in  $N$  Monte Carlo runs. The CRLB for this problem is  $\sigma_{CRLB}^2 = 10$ . The sample variance obtained from  $N = 100$  runs is  $\hat{\sigma}^2 = 7.811$ . Your best friend is concerned and tells you that you must have made a mistake somewhere. What is your answer to this? Give a quantitative justification for it — you can make any reasonable assumptions.

- 2-13 MLE from correlated measurements.** Given the three estimates of the scalar  $x$

$$z_i \triangleq \hat{x}_i = x + \tilde{x}_i \quad i = 1, 2, 3$$

with the estimation errors  $\tilde{x}_i$  jointly Gaussian, zero-mean, with

$$E[\tilde{x}_i \tilde{x}_j] = P_{ij} \quad i, j = 1, 2, 3$$

find

1. The MLE  $\hat{x}(z_1, z_2, z_3)$
2. The variance  $\sigma^2$  of the above MLE

**2-14 Measurement error variance.** We have a measuring device for which we are to ascertain that its measurement error variance is less than the borderline acceptable value of 100. You can assume that the errors are zero-mean.

We carry out  $N$  independent trials and the result is that the estimate of the variance is  $\hat{\sigma}^2 = 80$ .

For what  $N$  are you willing to risk your job by saying that there is a 5% (or less) probability to get such an estimate while the true value is  $\sigma^2 = 100$  (or higher)?

*Hint:* You can assume a Gaussian distribution for the error in the estimate of the variance and use a probability region about the borderline value.

**2-15 Estimation with correlated measurements.** Given the scalar random variables  $x_i$ ,  $i = 1, \dots, N + 1$ , with  $E x_i = \mu$  and  $\text{cov}[x_i, x_j] = \sigma^2 \rho^{|i-j|}$ , where  $|\rho| < 1$ .

1. Find the mean and variance of the following estimate of  $\mu$

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^N x_i$$

You can assume  $N \gg 1$ .

2. Is this a consistent estimator? Justify the answer.
3. To how many i.i.d. random variables with the same first two moments are the above equivalent (i.e., they yield the same variance) for  $\rho = 0.5$ ?

**2-16 Public opinion polls.**

1. Find the margin of error of a public opinion poll with  $N = 625$  subjects.
2. How many subjects are needed for a margin of error of 1%?



# Chapter 3

## LINEAR ESTIMATION IN STATIC SYSTEMS

### 3.1 INTRODUCTION

#### 3.1.1 Outline

This chapter presents the minimum mean square error (MMSE) estimation of Gaussian random vectors (Section 3.2) and the *linear* MMSE estimator for arbitrarily distributed random vectors (Section 3.3). The latter is the estimator constrained to have a linear form. The estimation of unknown constant vectors according to the least squares (LS) criterion is then discussed in Section 3.4, where both the batch and recursive versions are derived. These results are then applied to polynomial fitting in Section 3.5.

Section 3.6 presents the statistical tools for deciding what is the appropriate order of the polynomial when fitting a set of data points. This latter method is especially important because, in practice, the models — for instance, for the motion of targets — are not known *a priori* and have to be inferred from the data.

A realistic example that deals with the localization of a target — a source of radiated energy — based on bearings-only measurements from a platform is presented in Section 3.7. This illustrates the use of nonlinear LS to a practical problem.

#### 3.1.2 Linear Estimation in Static Systems — Summary of Objectives

Present

- MMSE estimation of Gaussian random vectors
- Linear MMSE estimator for arbitrarily distributed random vectors
- LS estimation of unknown constant vectors from *linear* observations

- batch form
- recursive form.

Apply the LS technique to

- Polynomial fitting
- Choice of order of the polynomial when fitting a set of data points.

Illustrate the use of *nonlinear* LS to a practical problem — target (source of radiated energy) localization from bearings-only measurements.

## 3.2 ESTIMATION OF GAUSSIAN RANDOM VECTORS

### 3.2.1 The Conditional Mean and Covariance for Gaussian Random Vectors

Consider two random vectors  $x$  and  $z$  that are *jointly* normally (Gaussian) distributed.

Define the *stacked vector*

$$y \triangleq \begin{bmatrix} x \\ z \end{bmatrix} \quad (3.2.1-1)$$

The notation

$$y \sim \mathcal{N}[\bar{y}, P_{yy}] \quad (3.2.1-2)$$

will indicate that the variable  $y$  is *normally (Gaussian) distributed* with mean

$$\bar{y} = \begin{bmatrix} \bar{x} \\ \bar{z} \end{bmatrix} \quad (3.2.1-3)$$

and covariance matrix (assumed nonsingular)

$$P_{yy} = \begin{bmatrix} P_{xx} & P_{xz} \\ P_{zx} & P_{zz} \end{bmatrix} \quad (3.2.1-4)$$

written in partitioned form, where  $\bar{x}$  is the mean of  $x$ ,

$$P_{xx} = E[(x - \bar{x})(x - \bar{x})'] \quad (3.2.1-5)$$

is its covariance, and

$$P_{xz} = E[(x - \bar{x})(z - \bar{z})'] = P'_{zx} \quad (3.2.1-6)$$

is the covariance between  $x$  and  $z$  (also called *cross-covariance*), and so on.

In the above,  $x$  is the random variable to be estimated and  $z$  is the *measurement* or the *observation*. As shown in (2.4.1-7), the *estimate* of the random variable  $x$  in terms of  $z$  according to the *minimum mean square*

**error (MMSE)** criterion — the **MMSE estimator** — is the *conditional mean* of  $x$  given  $z$ .

For  $x$  and  $z$  *jointly Gaussian*, as assumed in (3.2.1-2), the **conditional mean** is

$$\hat{x} \triangleq E[x|z] = \bar{x} + P_{xz}P_{zz}^{-1}(z - \bar{z}) \quad (3.2.1-7)$$

and the corresponding **conditional covariance matrix** is

$$P_{xx|z} \triangleq E[(x - \hat{x})(x - \hat{x})'|z] = P_{xx} - P_{xz}P_{zz}^{-1}P_{zx} \quad (3.2.1-8)$$

(see Subsection 1.4.14 for proof).

This follows from the fact that the conditional pdf of  $x$  given  $z$  is Gaussian with mean (3.2.1-7) and covariance (3.2.1-8).

Note that the optimal estimator (in the MMSE sense) of  $x$  in terms of  $z$  is a *linear function* of  $z$ . This is a consequence of the Gaussian assumption.

Another important property specific to this Gaussian problem is that the *conditional covariance* (3.2.1-8), which measures the “quality” of the estimate, is *independent of the observation  $z$* .

## 3.2.2 Estimation of Gaussian Random Vectors — Summary

The *MMSE estimate* — the *conditional mean* — of a *Gaussian* random vector in terms of another *Gaussian* random vector (the measurement) is a *linear combination* of

- The prior (unconditional) mean of the variable to be estimated;
- The difference between the measurement and its prior mean.

The *conditional covariance* of one *Gaussian* random vector given another *Gaussian* random vector (the measurement) is *independent* of the measurement.

Both of the above properties hinge strictly on the assumption that the two random vectors under consideration are *jointly Gaussian*.

## 3.3 LINEAR MINIMUM MEAN SQUARE ERROR ESTIMATION

### 3.3.1 The Principle of Orthogonality

The *minimum mean square error (MMSE) estimate* of a random variable  $x$  in terms of another random variable  $z$  is, according to (2.4.1-7), the conditional mean  $E[x|z]$ .

In many problems the distributional information needed for the evaluation of the conditional mean is not available. Furthermore, even if it were available, the evaluation of the conditional mean could be prohibitively complicated.

In view of this, a method that (1) is simple — yields the estimate as a linear function of the observation(s) and (2) requires little information — only first and second moments, is highly desirable. Such a method, called *linear MMSE estimation*, relies on the *principle of orthogonality*:

The best linear estimate (in the sense of MMSE) of a random variable in terms of another random variable — the observation(s) — is such that

1. The estimate is unbiased — the estimation error has mean zero, and
2. The estimation error is uncorrelated from the observation(s);

that is, they are *orthogonal*.

### Linear MMSE Estimation for Zero-Mean Random Variables

The linear MMSE estimation can be formulated in terms of a (normed linear) space of random variables as follows.

The set of real-valued scalar *zero-mean random variables*  $z_i$ ,  $i = 1, \dots, n$ , can be considered as *vectors in an abstract vector space* or *linear space*. Such a space is closed under addition of its elements and multiplication by scalars — the linear combination of two random variables is another element in this space.

A (complete) vector space in which one defines an *inner product* is a Hilbert space. The inner product that can be defined is

$$\langle z_i, z_k \rangle = E[z_i z_k] \quad (3.3.1-1)$$

Since the random variables under consideration are zero mean, it is clear that

$$\langle z_i, z_i \rangle = E[z_i^2] = \|z_i\|^2 \quad (3.3.1-2)$$

satisfies the properties of a *norm* and can be taken as such.

With this definition of the norm, *linear dependence* is defined by stating that the norm of a linear combination of vectors is zero

$$E \left[ \left( \sum_{i=1}^m \alpha_i z_i \right)^2 \right] = 0 \quad (3.3.1-3)$$

If  $\alpha_1 \neq 0$ , then  $z_1$  is a linear combination of  $z_2, \dots, z_m$ , namely,

$$z_1 = -\frac{1}{\alpha_1} \sum_{i=2}^m \alpha_i z_i \quad (3.3.1-4)$$

that is, it is an element of the *subspace* spanned by  $z_2, \dots, z_m$ . Two vectors are *orthogonal*, denoted as  $z_i \perp z_k$ , if and only if

$$\langle z_i, z_k \rangle = 0 \quad (3.3.1-5)$$

which is equivalent to these zero-mean random variables being *uncorrelated*.

The **linear MMSE estimator** of a zero-mean random variable  $x$  in terms of  $z_i$ ,  $i = 1, \dots, n$ , is given by

$$\hat{x} = \sum_{i=1}^n \beta_i z_i \quad (3.3.1-6)$$

and has to be such that the norm of the **estimation error**

$$\tilde{x} \triangleq x - \hat{x} \quad (3.3.1-7)$$

is minimum. The linear MMSE estimate is denoted also by a circumflex (“hat”), even though it is not the conditional mean as in (3.2.1-7).

Thus the norm of the estimation error

$$\|\tilde{x}\|^2 = E[(x - \hat{x})^2] = E\left[\left(x - \sum_{i=1}^n \beta_i z_i\right)^2\right] \quad (3.3.1-8)$$

will have to be minimized with respect to  $\beta_i$ ,  $i = 1, \dots, n$ .

Setting the derivative of (3.3.1-8) with respect to  $\beta_k$  to zero

$$-\frac{1}{2} \frac{\partial}{\partial \beta_k} \|\tilde{x}\|^2 = E\left[\left(x - \sum_{i=1}^n \beta_i z_i\right) z_k\right] = E[\tilde{x} z_k] = \langle \tilde{x}, z_k \rangle = 0 \quad k = 1, \dots, n \quad (3.3.1-9)$$

is seen to be equivalent to requiring the following *orthogonality* property:

$$\tilde{x} \perp z_k \quad \forall k \quad (3.3.1-10)$$

This is the **principle of orthogonality**: In order for the error to have minimum norm, it has to be *orthogonal to the observations*. This is equivalent to stating that the estimate  $\hat{x}$  has to be the **orthogonal projection** of  $x$  into the space spanned by the observations, as illustrated in Fig. 3.3.1-1.

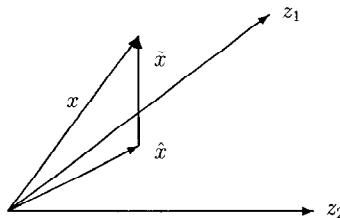


Figure 3.3.1-1: Orthogonal projection of random variable  $x$  into the subspace of  $\{z_1, z_2\}$ .

### Linear MMSE Estimation for Nonzero-Mean Random Variables

For a random variable  $x$  with nonzero mean  $\bar{x}$ , the best linear estimator is of the form (an affine function)

$$\hat{x} = \beta_0 + \sum_{i=1}^n \beta_i z_i \quad (3.3.1-11)$$

Since the MSE is the sum of the square of the mean and the variance

$$E[\tilde{x}^2] = (E[\tilde{x}])^2 + \text{var}(\tilde{x}) \quad (3.3.1-12)$$

in order to minimize it, the estimate should have the **unbiasedness property**:

$$E[\tilde{x}] = 0 \quad (3.3.1-13)$$

It can be easily shown that this follows from the fact that  $\beta_0$  enters only into the mean of the error (3.3.1-13), whose minimum norm is zero. Equation (3.3.1-13) yields

$$\beta_0 = \bar{x} - \sum_{i=1}^n \beta_i \bar{z}_i \quad (3.3.1-14)$$

where

$$\bar{z}_i = E[z_i] \quad (3.3.1-15)$$

Inserting (3.3.1-14) into (3.3.1-11) leads to

$$\hat{x} = \bar{x} + \sum_{i=1}^n \beta_i (z_i - \bar{z}_i) \quad (3.3.1-16)$$

The error corresponding to the estimate (3.3.1-16) is

$$\begin{aligned} \tilde{x} &\triangleq x - \hat{x} \\ &= x - \bar{x} - \sum_{i=1}^n \beta_i (z_i - \bar{z}_i) \end{aligned} \quad (3.3.1-17)$$

This has transformed the nonzero-mean case into the zero-mean case.

The orthogonality principle (3.3.1-10) then yields the coefficients  $\beta_i$  from the following equations:

$$\langle \tilde{x}, z_k \rangle = E[\tilde{x} z_k] = E\left[\left[x - \bar{x} - \sum_{i=1}^n \beta_i (z_i - \bar{z}_i)\right] z_k\right] = 0 \quad k = 1, \dots, n \quad (3.3.1-18)$$

The estimator (3.3.1-16) is also known as the **best linear unbiased estimator**.

### 3.3.2 Linear MMSE Estimation for Vector Random Variables

Consider the vector-valued random variables  $x$  and  $z$ , which are not necessarily Gaussian or zero-mean. The “best linear” estimate of  $x$  in terms of  $z$  is obtained as follows. The criterion for “best” is the MMSE; that is, find the estimator

$$\hat{x} = Az + b \quad (3.3.2-1)$$

that minimizes the *scalar MSE criterion*, which in the multidimensional case is the *expected value of the squared norm* of the estimation error,

$$J \triangleq E[(x - \hat{x})'(x - \hat{x})] \quad (3.3.2-2)$$

According to the previous discussion, the *linear MMSE estimator* is such that the estimation error

$$\tilde{x} = x - \hat{x} \quad (3.3.2-3)$$

is *zero-mean* (the estimate is unbiased) and *orthogonal to the observation  $z$* . In other words, the estimate  $\hat{x}$  is the orthogonal projection of the vector  $x$  into the space spanned by the (random components of the) observation vector  $z$ . The unbiasedness requirement (3.3.1-13) is

$$E[\tilde{x}] = \bar{x} - (A\bar{z} + b) = 0 \quad (3.3.2-4)$$

and it yields

$$b = \bar{x} - A\bar{z} \quad (3.3.2-5)$$

The estimation error is then

$$\tilde{x} = x - \bar{x} - A(z - \bar{z}) \quad (3.3.2-6)$$

The orthogonality requirement is, in the multidimensional case, that *each component* of  $\tilde{x}$  be orthogonal to *each component* of  $z$ .

The orthogonality requirement can thus be written as

$$E[\tilde{x}z'] = E\{[x - \bar{x} - A(z - \bar{z})]z'\} \quad (3.3.2-7)$$

$$= E\{[x - \bar{x} - A(z - \bar{z})](z - \bar{z})'\} = P_{xz} - AP_{zz} = 0 \quad (3.3.2-8)$$

The subtraction of  $\bar{z}$  from  $z$  in the transition from (3.3.2-7) to (3.3.2-8) could be done in view of the property (3.3.2-4) that  $\tilde{x}$  is zero mean.

The solution for the *weighting matrix  $A$*  is thus

$$A = P_{xz}P_{zz}^{-1} \quad (3.3.2-9)$$

The existence of the above requires the invertibility of  $P_{zz}$ , i.e., no linear dependence between the measurements (or, equivalently, no redundant measurements). Combining (3.3.2-5) and (3.3.2-9) yields the expression of the *linear MMSE estimator for the multidimensional case* as

$$\boxed{\hat{x} = \bar{x} + P_{xz}P_{zz}^{-1}(z - \bar{z})} \quad (3.3.2-10)$$

which is *identical* to the conditional mean (3.2.1-7) from the Gaussian case.

The **matrix MSE** corresponding to (3.3.2-10) is given by

$$E[\tilde{x}\tilde{x}'] = E[[x - \bar{x} - P_{xz}P_{zz}^{-1}(z - \bar{z})][x - \bar{x} - P_{xz}P_{zz}^{-1}(z - \bar{z})]'] \quad (3.3.2-11)$$

This becomes, after simple manipulations,

$$\boxed{E[\tilde{x}\tilde{x}'] = P_{xx} - P_{xz}P_{zz}^{-1}P_{zx} = P_{xx|z}} \quad (3.3.2-12)$$

that is, *identical* expression to the conditional covariance (3.2.1-8) in the Gaussian case. Note, however, that, strictly speaking, the matrix MSE (3.3.2-12) is not a covariance matrix since (3.3.2-10) is not the conditional mean.

Equations (3.3.2-10) and (3.3.2-12) are the **fundamental equations of linear estimation**.

### Remarks

Note the distinction between the scalar MSE criterion (3.3.2-2), an inner product, and the matrix MSE (3.3.2-12), an outer product. The **matrix MSE** is sometimes called, with abuse of language, a **covariance matrix**.

From the above derivations it follows that

- the *best estimator* (in the MMSE sense) for *Gaussian random variables* is identical to
- the *best linear estimator* for *arbitrarily distributed random variables with the same first- and second-order moments*.

The linear estimator (3.3.2-10) is the overall best if the random variables are Gaussian; otherwise, it is only the *best within the class of linear estimators*.

### Gaussian Assumption as “Worst Case” in MMSE Estimation

The following statement can be made: From the point of view of MMSE estimation, one can view the **Gaussian assumption as the worst case**:

- If the random variables are Gaussian, the minimum achievable matrix MSE is (3.2.1-8).
- If they are not Gaussian, but with the same first two moments, one can achieve (with the linear estimator) the matrix MSE (3.3.2-12), which is the same as (3.2.1-8).

However, in the non-Gaussian case,

The conditional mean (if one can compute it), being the *absolute best* as opposed to the *best within the class of linear estimators*, would give a matrix MSE (covariance matrix in this case) *less or equal* to (3.3.2-12).

## On the Terminology

The LMMSE estimator is also referred to in the literature as the *least mean square (LMS)* or *minimum variance (MV)* or *least squares (LS)*.

### 3.3.3 Linear MMSE Estimation — Summary

The *linear MMSE estimator* of one random vector in terms of another random vector (the measurement) is such that the estimation error is

1. Zero-mean (the estimate is unbiased)
2. Uncorrelated from the measurements

These two properties imply that the error is orthogonal to the measurements. This is the *principle of orthogonality*.

The expression of the *linear MMSE estimator* is identical to the expression of the *conditional mean* of Gaussian random vectors if they have the same first two moments.

Similarly, the *matrix MSE* associated with the LMMSE estimator has the same expression as the *conditional covariance* in the Gaussian case.

The *linear MMSE estimator* is

1. The *overall best* if the random variables are *Gaussian*
2. The *best* within the class of *linear* estimators otherwise (i.e., there is room for improvement in this case<sup>1</sup>)

## 3.4 LEAST SQUARES ESTIMATION

### 3.4.1 The Batch LS Estimation

In the *linear least squares (LS)* problem it is desired to estimate the  $n_x$ -vector  $x$ , modeled as an *unknown constant*, from the linear observations ( $n_z$ -vectors)

$$z(i) = H(i)x + w(i) \quad i = 1, \dots, k \quad (3.4.1-1)$$

such as to minimize the quadratic error

$$J(k) = \sum_{i=1}^k [z(i) - H(i)x]' R(i)^{-1} [z(i) - H(i)x] \quad (3.4.1-2)$$

weighted with the inverses of the positive definite matrices  $R(i)$ .

The above can be rewritten in a compact form as

$$J(k) = [z^k - H^k x]' (R^k)^{-1} [z^k - H^k x] \quad (3.4.1-3)$$

---

<sup>1</sup>If one is willing and capable of “going nonlinear” to obtain the optimal (nonlinear) estimator.

where

$$z^k = \begin{bmatrix} z(1) \\ \vdots \\ z(k) \end{bmatrix} \quad (3.4.1-4)$$

is the **stacked vector** of measurements (of dimension  $kn_z \times 1$ ),

$$H^k = \begin{bmatrix} H(1) \\ \vdots \\ H(k) \end{bmatrix} \quad (3.4.1-5)$$

is the **stacked measurement matrix** (of dimension  $kn_z \times n_x$ ),

$$w^k = \begin{bmatrix} w(1) \\ \vdots \\ w(k) \end{bmatrix} \quad (3.4.1-6)$$

is the stacked vector of the measurement errors, and

$$R^k = \begin{bmatrix} R(1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R(k) \end{bmatrix} = \text{diag}[R(i)] \quad (3.4.1-7)$$

Note that the matrix (3.4.1-7) is a block-diagonal positive definite  $kn_z \times kn_z$  matrix. If in (3.4.1-1) there is a nonlinear function of the unknown vector  $x$ , then one has a **nonlinear LS** problem (see Subsection 3.4.4 and Section 3.7).

The **LS estimator** that minimizes (3.4.1-3) is obtained by setting its gradient with respect to  $x$  to zero

$$\nabla_x J(k) = -2H^{k'}(R^k)^{-1}[z^k - H^k x] = 0 \quad (3.4.1-8)$$

which yields

$$\hat{x}(k) = [H^{k'}(R^k)^{-1}H^k]^{-1}H^{k'}(R^k)^{-1}z^k \quad (3.4.1-9)$$

assuming the required inverse exists.

It can be easily shown that since  $R^k$ , defined in (3.4.1-7), is positive definite, the Hessian of (3.4.1-3) with respect to  $x$  is positive definite, and consequently the extremum point (3.4.1-9) is a minimum.

Note that (3.4.1-9) is a **batch estimator** — the entire data have to be processed simultaneously for every  $k$ .

### Remark

In this approach,  $x$  is an unknown constant. The estimate  $\hat{x}(k)$  is a random variable if the disturbances (measurement errors)  $w(i)$  are modeled as random.

## Relationship to the Maximum Likelihood (ML) Estimator

If the measurement errors  $w(i)$  are *independent Gaussian* random variables with mean zero and covariance  $R(i)$ , then minimizing the LS criterion (3.4.1-2) is equivalent to maximizing the likelihood function

$$\begin{aligned}\Lambda_k(x) &= p[z^k|x] = \prod_{i=1}^k p[z(i)|x] \\ &= c e^{-\frac{1}{2} \sum_{i=1}^k [z(i) - H(i)x]' R(i)^{-1} [z(i) - H(i)x]}\end{aligned}\quad (3.4.1-10)$$

that is, the LS and ML estimators coincide in this case.

The LS criterion (3.4.1-2) implicitly assumes that  $w(i)$  are independent and zero mean with covariance  $R(i)$ , and it leads to the minimization of the sum of their weighted norms. Since this is equivalent to the maximization of the likelihood function under the additional Gaussian assumption, the LS is clearly a “disguised” ML technique.

## Properties of the LS Estimator

With the assumption that  $w(i)$  are uncorrelated, zero-mean random variables with covariance  $R(i)$ , but without any further distributional assumptions, the LS estimator (3.4.1-9) is *unbiased*, that is,

$$E[\hat{x}(k)] = [H^{k'}(R^k)^{-1}H^k]^{-1}H^{k'}(R^k)^{-1}E[H^kx + w^k] = x \quad (3.4.1-11)$$

The estimation error is

$$\tilde{x}(k) = x - \hat{x}(k) = -[H^{k'}(R^k)^{-1}H^k]^{-1}H^{k'}(R^k)^{-1}w^k \quad (3.4.1-12)$$

Thus, the **covariance matrix of the LS estimator** is

$$\begin{aligned}P(k) &\triangleq E[\{\hat{x}(k) - E[\hat{x}(k)]\}\{\hat{x}(k) - E[\hat{x}(k)]\}'] \\ &= E[\tilde{x}(k)[\hat{x}(k) - x]'] \\ &= E[\tilde{x}(k)\tilde{x}(k)'] \\ &= [H^{k'}(R^k)^{-1}H^k]^{-1}H^{k'}(R^k)^{-1}R^k(R^k)^{-1}H^k[H^{k'}(R^k)^{-1}H^k]^{-1}\end{aligned}\quad (3.4.1-13)$$

where in the last line above use was made of (3.4.1-12) and the fact that based on (3.4.1-7) one has

$$E[w^k w^{k'}] = R^k \quad (3.4.1-14)$$

Equation (3.4.1-13) yields, after cancellations,

$$P(k) = [H^{k'}(R^k)^{-1}H^k]^{-1} \quad (3.4.1-15)$$

Note that when carrying out the expectation in (3.4.1-11) and (3.4.1-13) it is over  $w^k$ .

## Existence of the Solution — Parameter Observability

The *existence of the inverse* of  $H'R^{-1}H$  required in (3.4.1-9) is equivalent to having the covariance of the error (3.4.1-15) *finite*. This amounts to requiring the parameter  $x$  to be **observable** — that is, it can be estimated from the observations. (See also problem 3-3.)

Small eigenvalues of  $H'R^{-1}H$  (near-singularity) which lead to large eigenvalues in (3.4.1-15) are an indication of **marginal (poor) observability**. The corresponding eigenvector(s) give the direction(s) in the parameter space where one has poor observability [Ham83].

### 3.4.2 The Recursive LS Estimator

A useful feature of the LS estimator (3.4.1-9) is that it can be rewritten in recursive form (i.e., suitable for sequential rather than batch processing). In this case,  $k$  is interpreted as “discrete time.”

When  $z(k+1)$  is obtained, one can write the following partitioned forms

$$z^{k+1} = \begin{bmatrix} z^k \\ z(k+1) \end{bmatrix} \quad (3.4.2-1)$$

$$H^{k+1} = \begin{bmatrix} H^k \\ H(k+1) \end{bmatrix} \quad (3.4.2-2)$$

$$w^{k+1} = \begin{bmatrix} w^k \\ w(k+1) \end{bmatrix} \quad (3.4.2-3)$$

$$R^{k+1} = \begin{bmatrix} R^k & 0 \\ 0 & R(k+1) \end{bmatrix} \quad (3.4.2-4)$$

### The Recursion for the Inverse Covariance

Expression (3.4.1-15) at  $k+1$  can be expressed recursively as

$$\begin{aligned} P(k+1)^{-1} &= H^{k+1'}(R^{k+1})^{-1}H^{k+1} \\ &= \left[ H^{k'} \ H(k+1)' \right] \left[ \begin{array}{cc} R^k & 0 \\ 0 & R(k+1) \end{array} \right]^{-1} \left[ \begin{array}{c} H^k \\ H(k+1) \end{array} \right] \\ &= H^{k'}(R^k)^{-1}H^k + H(k+1)'R(k+1)^{-1}H(k+1) \end{aligned} \quad (3.4.2-5)$$

or

$$\boxed{P(k+1)^{-1} = P(k)^{-1} + H(k+1)'R(k+1)^{-1}H(k+1)} \quad (3.4.2-6)$$

This can be interpreted as follows: The **information** (in the sense of Fisher, that is, the inverse covariance) at  $k+1$  equals the sum of the information

at  $k$  and the new information about  $x$  obtained from  $z(k+1)$ . The information is *additive* here because of the following:

1. The problem is static — the parameter is fixed.
2. The observations are modeled as *independent*.

Using the matrix inversion lemma (1.3.3-11), (3.4.2-6) can be rewritten as

$$\begin{aligned} P(k+1) &= [P(k)^{-1} + H(k+1)'R(k+1)^{-1}H(k+1)]^{-1} \\ &= P(k) - P(k)H(k+1)'[H(k+1)P(k)H(k+1)' \\ &\quad + R(k+1)]^{-1}H(k+1)P(k) \end{aligned} \tag{3.4.2-7}$$

## The Residual Covariance and the Update Gain

Denote the matrices

$$S(k+1) \triangleq H(k+1)P(k)H(k+1)' + R(k+1) \tag{3.4.2-8}$$

$$W(k+1) \triangleq P(k)H(k+1)'S(k+1)^{-1} \tag{3.4.2-9}$$

which, as will be seen later, have the interpretations of *covariance of the residual* and parameter *update gain*, respectively.

## The Recursion for the Covariance

With (3.4.2-8) and (3.4.2-9), recursion (3.4.2-7) can be rewritten more compactly as

$$P(k+1) = [I - W(k+1)H(k+1)]P(k) \tag{3.4.2-10}$$

that is,

$$P(k+1) = P(k) - W(k+1)S(k+1)W(k+1)' \tag{3.4.2-11}$$

which is an alternative to (3.4.2-7).

## Alternative Expression for the Gain

Using (3.4.2-7), one has the following identity

$$\begin{aligned} P(k+1)H(k+1)'R(k+1)^{-1} &= \{P(k)H(k+1)' - P(k)H(k+1)' \\ &\quad \cdot [H(k+1)P(k)H(k+1)' + R(k+1)]^{-1} \\ &\quad \cdot H(k+1)P(k)H(k+1)'\}R(k+1)^{-1} \\ &= P(k)H(k+1)' \\ &\quad \cdot [H(k+1)P(k)H(k+1)' + R(k+1)]^{-1} \end{aligned}$$

$$\begin{aligned}
 & \cdot \{H(k+1)P(k)H(k+1)' + R(k+1) \\
 & - H(k+1)P(k)H(k+1)'\} R(k+1)^{-1} \\
 & = P(k)H(k+1)' S(k+1)^{-1} \\
 & = W(k+1)
 \end{aligned} \tag{3.4.2-12}$$

This gives an alternative expression for the update gain (3.4.2-9) as

$$W(k+1) = P(k+1)H(k+1)'R(k+1)^{-1} \tag{3.4.2-13}$$

### The Recursion for the Estimate

The batch estimation equation (3.4.1-9) for  $k+1$  is rewritten as

$$\begin{aligned}
 \hat{x}(k+1) &= P(k+1)H^{k+1\prime}(R^{k+1})^{-1}z^{k+1} \\
 &= P(k+1) \begin{bmatrix} H^k & H(k+1)' \end{bmatrix} \begin{bmatrix} R^k & 0 \\ 0 & R(k+1) \end{bmatrix}^{-1} \begin{bmatrix} z^k \\ z(k+1) \end{bmatrix} \\
 &= P(k+1)H^k(R^k)^{-1}z^k + P(k+1)H(k+1)'R(k+1)^{-1}z(k+1) \\
 &= [I - W(k+1)H(k+1)]P(k)H^k(R^k)^{-1}z^k + W(k+1)z(k+1) \\
 &= [I - W(k+1)H(k+1)]\hat{x}(k) + W(k+1)z(k+1)
 \end{aligned} \tag{3.4.2-14}$$

where (3.4.1-9), (3.4.1-15), (3.4.2-10) and (3.4.2-13) were used.

The above is the **recursive parameter estimate updating equation** — the **recursive LS estimator**, written as

$$\hat{x}(k+1) = \hat{x}(k) + W(k+1)[z(k+1) - H(k+1)\hat{x}(k)] \tag{3.4.2-15}$$

The new (updated) estimate  $\hat{x}(k+1)$  is therefore equal to the previous one plus a **correction term**. This correction term consists of the **gain**  $W(k+1)$  multiplying the **residual** — the difference between the observation  $z(k+1)$  and the **predicted value** of this observation from the previous  $k$  measurements.

### Remark

Since this is a recursive scheme, **initialization** is required. This can be done by using a batch technique on a small number of initial measurements or by using an “a priori” initial estimate and an associated covariance.

### The Residual Covariance

It can be easily shown that  $S(k+1)$  defined in (3.4.2-8) is the **covariance of the residual** from (3.4.2-15), that is,

$$E[[z(k+1) - H(k+1)\hat{x}(k)][z(k+1) - H(k+1)\hat{x}(k)]'] = S(k+1) \tag{3.4.2-16}$$

See also problem 3-6.

### 3.4.3 Examples and Incorporation of Prior Information

#### The Sample Mean

Consider noisy observations on a constant scalar  $x$

$$z(i) = x + w(i) \quad i = 1, \dots, k \quad (3.4.3-1)$$

For the batch LS formulation, one has

$$H^k = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \quad (3.4.3-2)$$

a  $k$ -dimensional vector, and let

$$R^k = I\sigma^2 \quad (3.4.3-3)$$

where  $I$  is the  $k \times k$  identity matrix.

Then, using (3.4.1-9) with (3.4.3-2) and (3.4.3-3) from above, one has the LS estimate

$$\begin{aligned} \hat{x}(k) &= [H^{k'}(R^k)^{-1}H^k]^{-1}H^{k'}(R^k)^{-1}z^k \\ &= \left\{ [1 \ \cdots \ 1] (I\sigma^2)^{-1} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \right\}^{-1} [1 \ \cdots \ 1] (I\sigma^2)^{-1} \begin{bmatrix} z(1) \\ \vdots \\ z(k) \end{bmatrix} \\ &= \frac{1}{k} \sum_{i=1}^k z(i) \end{aligned} \quad (3.4.3-4)$$

that is, the **sample mean**. The variance of this estimate, assuming  $w(i)$  to be a sequence of independent and identically distributed random variables that are zero mean and with variance  $\sigma^2$ , follows from (3.4.1-15) as

$$P(k) = [H^{k'}(R^k)^{-1}H^k]^{-1} = \frac{\sigma^2}{k} \quad (3.4.3-5)$$

Note that this is the same result as obtained by direct evaluation in (2.6.3-1).

To obtain the recursive form of the LS estimation, using (3.4.2-8) and (3.4.2-9) one has

$$S(k+1) = H(k+1)P(k)H(k+1)' + R(k+1) = \frac{\sigma^2}{k} + \sigma^2 = \frac{k+1}{k}\sigma^2 \quad (3.4.3-6)$$

$$W(k+1) = P(k)H(k+1)'S(k+1)^{-1} = \frac{\sigma^2}{k} \left( \frac{k+1}{k} \sigma^2 \right)^{-1} = \frac{1}{k+1} \quad (3.4.3-7)$$

and thus, using (3.4.2-15)

$$\boxed{\hat{x}(k+1) = \hat{x}(k) + \frac{1}{k+1}[z(k+1) - \hat{x}(k)]} \quad (3.4.3-8)$$

The above recursion could have also been obtained directly from the batch expression by the following simple algebraic manipulation

$$\begin{aligned}\hat{x}(k+1) &= \frac{1}{k+1} \sum_{i=1}^{k+1} z(i) \\ &= \frac{1}{k+1} \left[ \sum_{i=1}^k z(i) + z(k+1) \right] \\ &= \frac{1}{k+1} [k\hat{x}(k) + z(k+1) - \hat{x}(k) + \hat{x}(k)] \\ &= \hat{x}(k) + \frac{1}{k+1}[z(k+1) - \hat{x}(k)] \quad k = 1, \dots\end{aligned} \quad (3.4.3-9)$$

with the initial condition  $\hat{x}(1) = z(1)$ .

### Estimation of the Mean with Prior Information

The previous example will be reconsidered with  $x$  now assumed a *random variable* with **prior information** consisting of the mean  $\bar{x}$  and the variance

$$P_{xx} = \sigma_0^2 \quad (3.4.3-10)$$

The estimation will be cast in the framework of the LMMSE estimation discussed in Section 3.3.

The measurements are as in (3.4.3-1) and  $w(i)$  are i.i.d. zero-mean, with variance  $\sigma^2$  and independent of  $x$ . Thus (dropping the superscript of  $z$ )

$$z = H^k x + w^k \quad (3.4.3-11)$$

Averaging over  $x$  and  $w$ , one has

$$\begin{aligned}P_{zz} &= E[(z - \bar{z})(z - \bar{z})'] = E[(H^k(x - \bar{x}) + w^k)][H^k(x - \bar{x}) + w^k]'] \\ &= H^k \sigma_0^2 H^{k'} + I \sigma^2\end{aligned} \quad (3.4.3-12)$$

and, similarly,

$$P_{xz} = E[(x - \bar{x})(z - \bar{z})'] = E[(x - \bar{x})[H^k(x - \bar{x}) + w^k]'] = \sigma_0^2 H^{k'} \quad (3.4.3-13)$$

The inverse of (3.4.3-12), using the matrix inversion lemma (1.3.3-12), is

$$\begin{aligned}P_{zz}^{-1} &= I \sigma^{-2} - \sigma^{-2} H^k (H^{k'} \sigma^{-2} H^k + \sigma_0^{-2})^{-1} H^{k'} \sigma^{-2} \\ &= I \sigma^{-2} - \frac{\sigma^{-4}}{k \sigma^{-2} + \sigma_0^{-2}} H^k H^{k'}\end{aligned} \quad (3.4.3-14)$$

The LMMSE estimate (3.3.2-10) of  $x$  is then

$$\begin{aligned}
 \hat{x}(k) &= \bar{x} + P_{xz}P_{zz}^{-1}(z - H^k\bar{x}) \\
 &= \bar{x} + \sigma_0^2 H^{k'} \left( I\sigma^{-2} - \frac{\sigma^{-4}}{k\sigma^{-2} + \sigma_0^{-2}} H^k H^{k'} \right) (z - H^k\bar{x}) \\
 &= \bar{x} + \sigma_0^2 \left( \sigma^{-2} - \frac{\sigma^{-4}}{k\sigma^{-2} + \sigma_0^{-2}} H^{k'} H^k \right) H^{k'} (z - H^k\bar{x}) \\
 &= \bar{x} + \sigma_0^2 \left( \sigma^{-2} - \frac{k\sigma^{-4}}{k\sigma^{-2} + \sigma_0^{-2}} \right) \sum_{i=1}^k [z(i) - \bar{x}] \\
 &= \bar{x} + \frac{\sigma^{-2}}{k\sigma^{-2} + \sigma_0^{-2}} \sum_{i=1}^k [z(i) - \bar{x}]
 \end{aligned} \tag{3.4.3-15}$$

or

$$\hat{x}(k) = \frac{\sigma_0^{-2}}{k\sigma^{-2} + \sigma_0^{-2}} \bar{x} + \frac{\sigma^{-2}}{k\sigma^{-2} + \sigma_0^{-2}} \sum_{i=1}^k z(i) \tag{3.4.3-16}$$

### Effect of Diffuse Prior

Note that if  $\sigma_0 \rightarrow \infty$ , the prior information becomes **diffuse**, which amounts to lack of prior information — this is the motivation of the term **noninformative**. In this case (3.4.3-16) becomes

$$\hat{x}(k) = \frac{1}{k} \sum_{i=1}^k z(i) \tag{3.4.3-17}$$

which is the LS or ML estimate, as expected. (See also problem 3-2.)

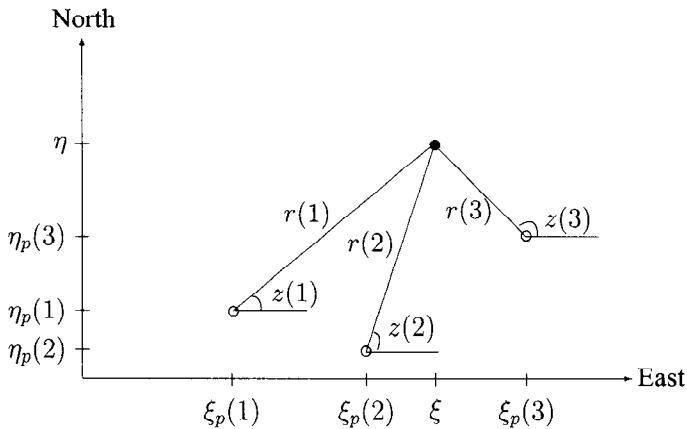
### 3.4.4 Nonlinear LS — An Example

In this section the use of the LS technique for estimating parameters from a set of nonlinear measurements is presented. The **Nonlinear Least Squares (NLS)** estimator is illustrated on the scenario in Figure 3.4.4-1, where **angle-only measurements** of a stationary target are obtained from a moving platform at different times.<sup>2</sup>

In Figure 3.4.4-1, the target position is denoted by  $x = [\xi \ \eta]$  and the sensor platform position at time  $i$  is denoted by  $x_p(i) = [\xi_p(i) \ \eta_p(i)]$ . The measurement  $z(i)$  at time  $i$  is given by

$$\begin{aligned}
 z(i) &= \tan^{-1} \left( \frac{\eta - \eta_p(i)}{\xi - \xi_p(i)} \right) + v(i) \\
 &= h(x, x_p(i)) + v(i) \quad i = 1, 2, 3
 \end{aligned} \tag{3.4.4-1}$$

<sup>2</sup>This problem is identical to the situation where simultaneous measurements are made from the same number of sensors. It is also known as **multilateration**, a generalized version of **triangulation**.



**Figure 3.4.4-1:** Sensor platform-target geometry for NLS estimation ( $\circ$  — platform,  $\bullet$  — target).

where  $v(i)$  is the zero-mean white Gaussian measurement noise with variance  $\sigma^2$ .

The stacked vector  $z$  of measurements obtained for  $i = 1, 2, 3$  is denoted by

$$\begin{aligned} z &= \begin{bmatrix} z(1) \\ z(2) \\ z(3) \end{bmatrix} \\ &= \mathbf{h}(x, x_p) + w \end{aligned} \quad (3.4.4-2)$$

where

$$\mathbf{h}(x) = \begin{bmatrix} h(x, x_p(1)) \\ h(x, x_p(2)) \\ h(x, x_p(3)) \end{bmatrix} \quad (3.4.4-3)$$

and  $v$  is the stacked vector of measurement noises.

## Iterated Least Squares Estimation

The **Iterated Least Squares (ILS)** estimator is a technique for iteratively improving the current estimate using the measurements until convergence (or up to a certain maximum number of iterations) based on the LS principle [Fletcher87, Blackman99].

Given the estimate  $\hat{x}_j^{\text{ILS}}$  at the end of iteration  $j$ , the updated ILS estimate  $\hat{x}_{j+1}^{\text{ILS}}$  is obtained as

$$\hat{x}_{j+1}^{\text{ILS}} = \hat{x}_j^{\text{ILS}} + (J'_j R^{-1} J_j)^{-1} J'_j R^{-1} [z - \mathbf{h}(\hat{x}_j^{\text{ILS}}, x_p)] \quad (3.4.4-4)$$

where

$$J_j = \left. \frac{\partial \mathbf{h}(x, x_p)}{\partial x} \right|_{x=\hat{x}_j^{\text{ILS}}} \quad (3.4.4-5)$$

is the Jacobian matrix and the measurement covariance matrix  $R$  is given by  $R = \text{diag}(\sigma^2, \sigma^2, \sigma^2)$ .

The ILS estimator progresses by applying a correction term to the current estimate based on the measurement residual. Note that (3.4.4-4) is similar to the recursive LS estimator obtained in (3.4.2-14).

The Jacobian matrix  $J_j$  is given by

$$J_j = \left[ \begin{array}{cc} \frac{\partial h(x, x_p(1))}{\partial \xi} & \frac{\partial h(x, x_p(1))}{\partial \eta} \\ \frac{\partial h(x, x_p(2))}{\partial \xi} & \frac{\partial h(x, x_p(2))}{\partial \eta} \\ \frac{\partial h(x, x_p(3))}{\partial \xi} & \frac{\partial h(x, x_p(3))}{\partial \eta} \end{array} \right] \Bigg|_{x=\hat{x}_j^{\text{ILS}}} \quad (3.4.4-6)$$

with

$$\begin{aligned} \frac{\partial h(x, x_p(i))}{\partial \xi} &= -\frac{\eta - \eta_p(i)}{(\xi - \xi_p(i))^2 + (\eta - \eta_p(i))^2} \\ \frac{\partial h(x, x_p(i))}{\partial \eta} &= \frac{\xi - \xi_p(i)}{(\xi - \xi_p(i))^2 + (\eta - \eta_p(i))^2} \end{aligned} \quad (3.4.4-7)$$

An initial estimate  $\hat{x}_0^{\text{ILS}}$  for the ILS estimator can be obtained from the intersection of any two of the three angle measurements. The iteration (3.4.4-4) is carried out until the result converges to within a certain accuracy.

The derivation of the ILS recursion is as follows. Using a first order series expansion about  $\hat{x}_j$  one has

$$z = \mathbf{h}(x, x_p) + w \approx \mathbf{h}(\hat{x}_j, x_p) + J_j(x - \hat{x}_j) + w \quad (3.4.4-8)$$

which can be rewritten (using now an equal sign, even though it is an approximation) as

$$z - \mathbf{h}(\hat{x}_j, x_p) = J_j(x - \hat{x}_j) + w \quad (3.4.4-9)$$

The above can be seen as a linear LS problem for  $x - \hat{x}_j$  with observations  $z - \mathbf{h}(\hat{x}_j, x_p)$ , the solution to which is

$$\hat{x}_{j+1} - \hat{x}_j = (J'_j R^{-1} J_j)^{-1} J'_j R^{-1} [z - \mathbf{h}(\hat{x}_j, x_p)] \quad (3.4.4-10)$$

which immediately yields (3.4.4-4).

The mean square error of the final estimate  $\hat{x}$  can be obtained (approximately) from the above with  $\hat{x}_{j+1} = \hat{x}$  and  $x$  instead of  $\hat{x}_j$ , i.e.,

$$\hat{x}_{j+1} - x = (J' R^{-1} J)^{-1} J' R^{-1} [z - \mathbf{h}(x, x_p)] = (J' R^{-1} J)^{-1} J' R^{-1} w \quad (3.4.4-11)$$

which then yields

$$E[(\hat{x}_{j+1} - x)(\hat{x}_{j+1} - x)'] = (J'R^{-1}J)^{-1} \quad (3.4.4-12)$$

where  $J$  is the Jacobian, evaluated at the final estimate. Note that the above is not a covariance but a **mean square error (MSE) matrix** because one cannot claim that the estimate is unbiased. Furthermore, it is an approximate one because it hinges on the accuracy of the expansion (3.4.4-8) on which the ILS is based.

### ML Estimation with Newton-Raphson Search

The ML estimate  $\hat{x}^{\text{ML}}$  is obtained by maximizing the likelihood function  $\Lambda(x)$  given by

$$\begin{aligned} \Lambda(x) &= \prod_{i=1}^3 p(z(i)|x) \\ &= \prod_{i=1}^3 ce^{-\frac{1}{2\sigma^2}[z(i)-h(x,x_p(i))]^2} \end{aligned} \quad (3.4.4-13)$$

where  $c$  is a normalizing constant. That is,

$$\begin{aligned} \hat{x}^{\text{ML}} &= \arg \max_x \Lambda(x) = \arg \min_x \{-\ln[\Lambda(x)]\} \\ &\triangleq \arg \min_x \lambda(x) \end{aligned} \quad (3.4.4-14)$$

where

$$\lambda(x) = \frac{1}{2\sigma^2} \sum_{i=1}^3 [z(i) - h(x, x_p(i))]^2 \quad (3.4.4-15)$$

Note that this is the same as the NLS problem formulation.

The minimization has to be carried out using a numerical search, for example, with the Newton-Raphson method, where the estimate  $\hat{x}_j^{\text{ML}}$  in the  $j$ th iteration is related to the next estimate  $\hat{x}_{j+1}^{\text{ML}}$  via

$$\hat{x}_{j+1}^{\text{ML}} = \hat{x}_j^{\text{ML}} - H_j^{-1} \nabla_x \lambda(x) \Big|_{x=\hat{x}_j^{\text{ML}}} \quad (3.4.4-16)$$

where the Hessian matrix  $H_j$  is given by

$$H_j = \left[ \begin{array}{cc} \frac{\partial^2 \lambda(x)}{\partial \xi^2} & \frac{\partial^2 \lambda(x)}{\partial \xi \partial \eta} \\ \frac{\partial^2 \lambda(x)}{\partial \xi \partial \eta} & \frac{\partial^2 \lambda(x)}{\partial \eta^2} \end{array} \right] \Big|_{x=\hat{x}_j^{\text{ML}}} \quad (3.4.4-17)$$

with

$$\begin{aligned}
 \frac{\partial \lambda(x)}{\partial \xi} &= \frac{1}{\sigma^2} \sum_{i=1}^3 [z(i) - h(x, x_p(i))] \frac{\partial h(x, x_p(i))}{\partial \xi} \\
 \frac{\partial \lambda(x)}{\partial \eta} &= \frac{1}{\sigma^2} \sum_{i=1}^3 [z(i) - h(x, x_p(i))] \frac{\partial h(x, x_p(i))}{\partial \eta} \\
 \frac{\partial^2 \lambda(x)}{\partial^2 \xi} &= \frac{1}{\sigma^2} \sum_{i=1}^3 [z(i) - h(x, x_p(i))] \frac{\partial^2 h(x, x_p(i))}{\partial^2 \xi} - \left( \frac{\partial h(x, x_p(i))}{\partial \xi} \right)^2 \\
 \frac{\partial^2 \lambda(x)}{\partial^2 \eta} &= \frac{1}{\sigma^2} \sum_{i=1}^3 [z(i) - h(x, x_p(i))] \frac{\partial^2 h(x, x_p(i))}{\partial^2 \eta} - \left( \frac{\partial h(x, x_p(i))}{\partial \eta} \right)^2 \\
 \frac{\partial^2 \lambda(x)}{\partial \xi \partial \eta} &= \frac{1}{\sigma^2} \sum_{i=1}^3 [z(i) - h(x, x_p(i))] \frac{\partial^2 h(x, x_p(i))}{\partial \xi \partial \eta} - \\
 &\quad \frac{\partial h(x, x_p(i))}{\partial \xi} \frac{\partial h(x, x_p(i))}{\partial \eta}
 \end{aligned} \tag{3.4.4-18}$$

and

$$\frac{\partial^2 h(x, x_p(i))}{\partial^2 \xi} = \frac{2(\eta - \eta_p(i))(\xi - \xi_p(i))}{[(\eta - \eta_p(i))^2 + (\xi - \xi_p(i))^2]^2} \tag{3.4.4-19}$$

$$\frac{\partial^2 h(x, x_p(i))}{\partial^2 \eta} = -\frac{2(\eta - \eta_p(i))(\xi - \xi_p(i))}{[(\eta - \eta_p(i))^2 + (\xi - \xi_p(i))^2]^2} \tag{3.4.4-20}$$

$$\frac{\partial^2 h(x, x_p(i))}{\partial \xi \partial \eta} = \frac{(\eta - \eta_p(i))^2 - (\xi - \xi_p(i))^2}{[(\eta - \eta_p(i))^2 + (\xi - \xi_p(i))^2]^2} \tag{3.4.4-21}$$

Note that the Hessian may lose its positive definiteness due to numerical approximations — it is impossible to ensure that the Newton step will be taken in a direction decreasing in value. To rectify this, quasi-Newton or variable metric methods, which use an approximate inverse Hessian to guarantee downhill progress toward the minimum in each Newton step, should be used in practice [Press92].

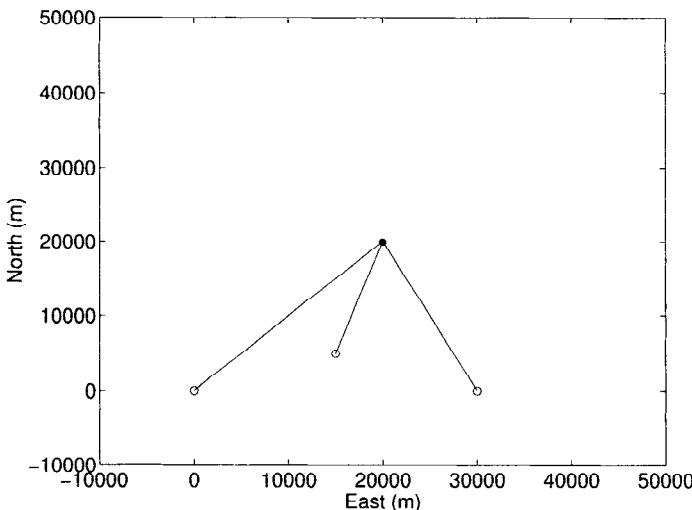
### The Cramer-Rao Lower Bound

The next step is to quantify the accuracy obtainable with an (unbiased) estimator. This is done by evaluating the CRLB (see Section 2.7.2)

$$E [(x - \hat{x})(x - \hat{x})'] \geq J^{-1} \tag{3.4.4-22}$$

where  $J$  is the Fisher information matrix (FIM) given by

$$\begin{aligned}
 J &= E \{ [\nabla_x \ln \Lambda(x)] [\nabla_x \ln \Lambda(x)]' \} \Big|_{x=x_{\text{true}}} \\
 &= E \{ [\nabla_x \lambda(x)] [\nabla_x \lambda(x)]' \} \Big|_{x=x_{\text{true}}}
 \end{aligned} \tag{3.4.4-23}$$



**Figure 3.4.4-2:** Scenario with good observability (○ — platform, • — target).

For this angle-only tracking problem, the FIM can be shown to be (see Section 3.7.4)

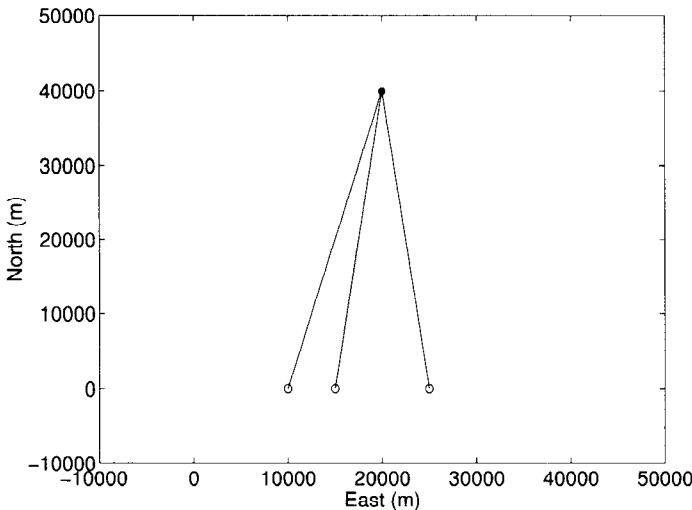
$$J = \frac{1}{\sigma^2} \sum_{i=1}^3 [\nabla_x h(x, x_p(i))] [\nabla_x h(x, x_p(i))]' \Big|_{x=x_{\text{true}}} \quad (3.4.4-24)$$

### Simulation Results

The estimation using the ILS and ML estimators is illustrated on two different scenarios, one with a “good” sensor-to-target geometry and the other with a “poor” geometry. That is, these two scenarios have good and poor observabilities, indicated by a low and a high CRLB values, respectively. The scenarios are shown in Figs. 3.4.4-2 and 3.4.4-3, respectively.

The RMS estimation errors obtained over 350 Monte Carlo runs, together with the corresponding CRLB values, are given in Table 3.4.4-1 and 3.4.4-2 for the above scenarios. The measurement standard deviation  $\sigma$  in both cases is  $1^\circ$ . The ILS and ML iterations terminate when the norm of the difference between two consecutive estimates is less than 0.5 m or when the number of iterations exceeds 50. Typically, the algorithms converge in about 5 iterations. Both algorithms are initialized with the intersection of the first and the third line-of-sight measurements.

It can be seen from Tables 3.4.4-1 and 3.4.4-2 that both estimators yield nearly identical results in the first scenario with good observability. However, in



**Figure 3.4.4-3:** Scenario with poor observability ( $\circ$  — platform,  $\bullet$  — target).

the second scenario with poor observability, the ILS estimator outperforms the ML estimator, which requires the Hessian for its calculations. This is because, with a poor geometry, the numerical evaluation of the Hessian is not accurate enough to guarantee its positive definiteness.

**Table 3.4.4-1:** Simulation results for the scenario in Figure 3.4.4-2.

Algorithm	$\xi$ (m)	$\bar{\xi}$ (m)	$\sigma_{\xi}^{\text{CRLB}}$ (m)	$\hat{\sigma}_{\xi}$ (m)	$\eta$ (m)	$\bar{\eta}$ (m)	$\sigma_{\eta}^{\text{CRLB}}$ (m)	$\hat{\sigma}_{\eta}$ (m)
ILS	20,000	20,000.9	244.7	268.9	20,000	20,001.1	494.7	512.2
ML	20,000	20,000.9	244.7	268.9	20,000	20,001.1	494.7	512.2

The estimation results from a single run on the first scenario are shown in Figs. 3.4.4-4 and 3.4.4-5 to illustrate the convergence of the estimates in both algorithms. It can be seen that the estimates converge to their final values within a few iterations.

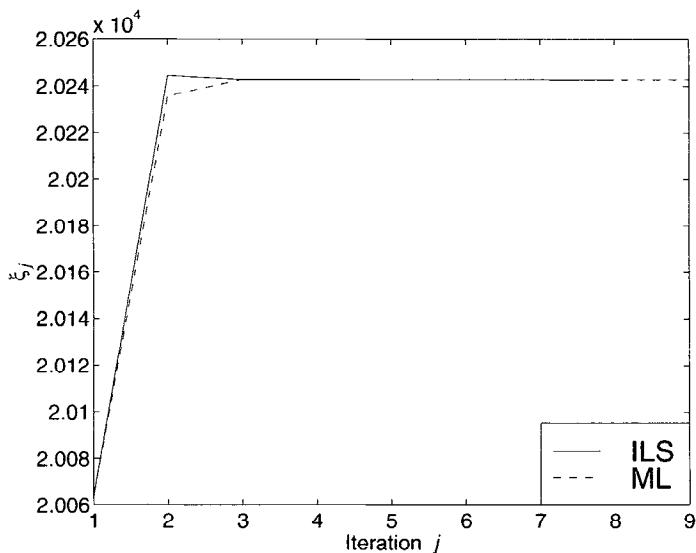


Figure 3.4.4-4: Convergence of iterative estimates  $\hat{\xi}_j$ .

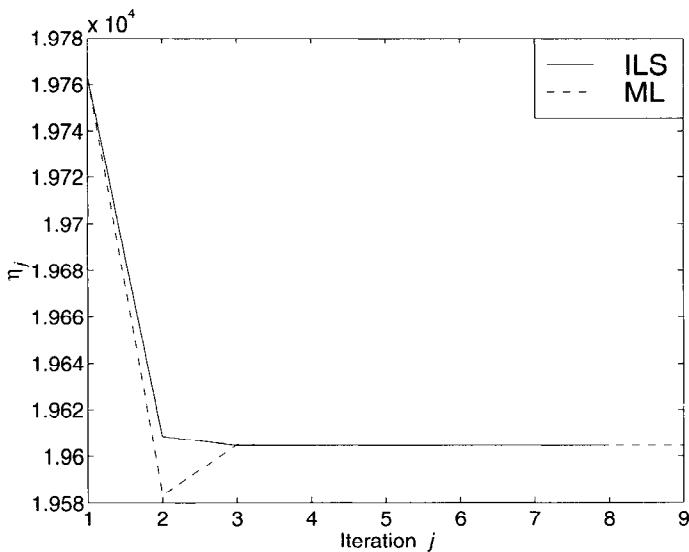


Figure 3.4.4-5: Convergence of iterative estimates  $\hat{\eta}_j$ .

**Table 3.4.4-2:** Simulation results for the scenario in Figure 3.4.4-3.

Algorithm	$\xi_{\text{true}}$ (m)	$\bar{\xi}$ (m)	$\sigma_{\xi}^{\text{CRLB}}$ (m)	$\hat{\sigma}_{\xi}$ (m)	$\eta_{\text{true}}$ (m)	$\bar{\eta}$ (m)	$\sigma_{\eta}^{\text{CRLB}}$ (m)	$\hat{\sigma}_{\eta}$ (m)
ILS	20,000	19,997.9	465.1	480.3	40,000	40,088.4	2671.4	2876.5
ML	20,000	20,003.5	465.1	520.2	40,000	40,065.8	2671.4	2928.1

### Geometric Dilution of Precision

The **Geometric Dilution of Precision (GDOP)** is defined as the degradation of the overall position estimate RMSE vs. (in this case) the crossrange accuracy of the sensor. That is,

$$\text{GDOP} = \frac{\text{Position RMSE}}{\text{Distance to target} \times \sigma} \quad (3.4.4-25)$$

Note that the angle standard deviation  $\sigma$  has to be in radians.

The GDOP is very sensitive to the sensor-target geometry — a higher value of GDOP indicates a poor geometry. In addition, the lower the GDOP the better the estimation algorithm. For the scenarios in Figs. 3.4.4-2 and 3.4.4-3, the GDOP values are in the range 1.0–1.3 and 3.9–4.0, respectively. As one would expect, the GDOP values are higher for the scenario with poor observability.

### 3.4.5 LS Estimation — Summary

The LS estimator based on a set (or a sequence) of linear measurements of an *unknown constant parameter* is a linear function of the *stacked measurement vector* — this is the *batch form* of the LS.

The LS criterion is really a disguised ML criterion under suitable Gaussian assumptions on the observation noises (zero mean, uncorrelated, and with the covariance given by the inverses of the LS criterion weighting matrices).

The LS estimator can be rewritten in a *recursive form*:

The estimate based on a given set of measurements is a *linear combination* of

1. the *previous estimate* (available prior to the latest measurement) and
2. the *latest measurement*.

The inverse of the covariance of the parameter estimate (the information matrix in the sense of Fisher) is also obtained *recursively* as the sum of:

1. the inverse covariance of the estimate prior to the latest measurement — the *old information* — and

2. the information about the parameter in the latest measurement — the ***new information***.

## 3.5 POLYNOMIAL FITTING

### 3.5.1 Fitting a First-Order Polynomial to Noisy Measurements

Assume that one measures, in the presence of additive noise, the position of an object moving in one dimension with constant velocity, that is,

$$z(i) = x_0 + \dot{x}_0 t_i + w(i) \quad i = 1, \dots \quad (3.5.1-1)$$

This motion is characterized by the unknown parameter

$$x = [x_0 \ \dot{x}_0]' \quad (3.5.1-2)$$

consisting of the object's *initial position and velocity* at  $t_0 = 0$ , the ***reference time***.

Equation (3.5.1-1), which is known in statistics as a first-order ***regression***, can be written as

$$z(i) = H(i)x + w(i) \quad (3.5.1-3)$$

where

$$H(i) = [1 \ t_i] \quad (3.5.1-4)$$

The problem of estimating the parameter  $x$  amounts to ***polynomial fitting***: in this case, fitting a first-order polynomial — a straight line — to a set of noisy measurements.

If the noises  $w(i)$  are independent, identically distributed, zero mean, and with variances  $\sigma^2$ , then according to (3.4.1-7), the covariance matrix of the stacked measurement noise vector is

$$R^k = I\sigma^2 \quad (3.5.1-5)$$

where  $I$  denotes the identity matrix of dimension  $k$ , not indicated for simplicity.

The batch solution is, noting that  $\sigma^2$  cancels,

$$\begin{aligned} \hat{x}(k) &= [H^k'(R^k)^{-1}H^k]^{-1}H^k'(R^k)^{-1}z^k = [H^k'(I\sigma^2)^{-1}H^k]^{-1}H^k'(I\sigma^2)^{-1}z^k \\ &= (H^k'H^k)^{-1}H^k'z^k \end{aligned} \quad (3.5.1-6)$$

Because of the special form of the noise covariance (3.5.1-5), the parameter estimate covariance can be written as

$$P(k) = [H^k'(R^k)^{-1}H^k]^{-1} = [H^k'(I\sigma^2)^{-1}H^k]^{-1} = (H^k'H^k)^{-1}\sigma^2 \quad (3.5.1-7)$$

The recursive solution is

$$\hat{x}(k+1) = \hat{x}(k) + W(k+1)[z(k+1) - H(k+1)\hat{x}(k)] \quad (3.5.1-8)$$

The **initialization** of recursion (3.5.1-8) is to be done using a batch estimate. This requires a minimum of two observations in (3.5.1-6), because otherwise the required inversion cannot be done. For  $k = 2$ , one has

$$\begin{aligned} P(2) &= (H^{2'} H^2)^{-1} \sigma^2 = \left\{ \begin{bmatrix} 1 & 1 \\ t_1 & t_2 \end{bmatrix} \begin{bmatrix} 1 & t_1 \\ 1 & t_2 \end{bmatrix} \right\}^{-1} \sigma^2 \\ &= \frac{1}{(t_2 - t_1)^2} \begin{bmatrix} t_1^2 + t_2^2 & -t_1 - t_2 \\ -t_1 - t_2 & 2 \end{bmatrix} \sigma^2 \end{aligned} \quad (3.5.1-9)$$

and

$$\begin{aligned} \hat{x}(2) &= P(2) H^{2'} \sigma^{-2} z^2 = \frac{1}{(t_2 - t_1)^2} \begin{bmatrix} t_1^2 + t_2^2 & -t_1 - t_2 \\ -t_1 - t_2 & 2 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ t_1 & t_2 \end{bmatrix} \begin{bmatrix} z(1) \\ z(2) \end{bmatrix} \\ &= \frac{1}{t_2 - t_1} \begin{bmatrix} z(1)t_2 - z(2)t_1 \\ z(2) - z(1) \end{bmatrix} \end{aligned} \quad (3.5.1-10)$$

The covariance matrix of the estimation error of the parameter vector (3.5.1-2) is, from  $k + 1$  measurements, given by

$$\begin{aligned} P(k+1) &= (H^{k+1'} H^{k+1})^{-1} \sigma^2 = \left\{ \begin{bmatrix} 1 & \cdots & 1 \\ t_1 & \cdots & t_{k+1} \end{bmatrix} \begin{bmatrix} 1 & t_1 \\ \vdots & \vdots \\ 1 & t_{k+1} \end{bmatrix} \right\}^{-1} \sigma^2 \\ &= \begin{bmatrix} s_0 & s_1 \\ s_1 & s_2 \end{bmatrix}^{-1} = \frac{1}{s_0 s_2 - s_1^2} \begin{bmatrix} s_2 & -s_1 \\ -s_1 & s_0 \end{bmatrix} \sigma^2 \end{aligned} \quad (3.5.1-11)$$

where

$$s_j \triangleq \sum_{i=1}^{k+1} (t_i)^j \quad j = 0, 1, 2 \quad (3.5.1-12)$$

denote compactly the three functions of the sampling times needed in (3.5.1-11).

Then, the gain (a two-dimensional vector) is obtained from (3.4.2-12) as

$$\begin{aligned} W(k+1) &= P(k+1) H(k+1)' R(k+1)^{-1} \\ &= \frac{1}{s_0 s_2 - s_1^2} \begin{bmatrix} s_2 - s_1 t_{k+1} \\ -s_1 + s_0 t_{k+1} \end{bmatrix} \end{aligned} \quad (3.5.1-13)$$

These expressions are general — the sampling times are arbitrary. They can be simplified for uniformly spaced samples, as will be seen next.

If the samples are uniformly spaced with sampling interval (period)  $T$ , that is,

$$t_i = iT \quad i = 1, 2, \dots \quad (3.5.1-14)$$

then the three functions in (3.5.1-12) have the following closed-form expressions:

$$s_0 = k + 1 \quad (3.5.1-15)$$

$$s_1 = \frac{(k+1)(k+2)}{2} T \quad (3.5.1-16)$$

$$s_2 = \frac{(k+1)(k+2)(2k+3)}{6} T^2 \quad (3.5.1-17)$$

In this case the gain (3.5.1-13) is

$$W(k+1) = \begin{bmatrix} -\frac{2}{(k+1)} \\ 6 \\ \hline (k+1)(k+2)T \end{bmatrix} \quad (3.5.1-18)$$

The explicit expression of the covariance (3.5.1-11) is

$$\begin{aligned} P(k+1) &= \frac{\sigma^2}{s_0 s_2 - s_1^2} \begin{bmatrix} s_2 & -s_1 \\ -s_1 & s_0 \end{bmatrix} \\ &= \frac{\sigma^2}{k(k+1)} \begin{bmatrix} 2(2k+3) & -\frac{6}{T} \\ -\frac{6}{T} & \frac{12}{(k+2)T^2} \end{bmatrix} \end{aligned} \quad (3.5.1-19)$$

### Remark

The gain (3.5.1-18) tends to zero as  $k \rightarrow \infty$  since the covariance  $P(k)$  tends to zero as more observations are made. The reason for the gain tending to zero is that *zero variance implies perfect estimate*, in which case *there is no more need to update the estimate*.

### Note

In statistics, regression is used to relate one variable,  $z$ , to a set of variables,  $y_j$

$$z(i) = \sum_{j=1}^n a_j y_j(i) + w(i) \quad (3.5.1-20)$$

with the variance of  $w(i)$  assumed, in general, *unknown*, to be estimated together with the coefficients  $a_j$ . In the case discussed above we had  $n = 2$ ,  $y_1(i) = 1$ ,  $y_2(i) = t_i$ ,  $a_1 = x_0$ ,  $a_2 = \dot{x}_0$ , and the variance of  $w(i)$  was assumed to be known. This last assumption is reasonable if we know that the measurements are made with a sensor with known accuracy, which is the case in (most of the) engineering problems.

### 3.5.2 Fitting a General Polynomial to a Set of Noisy Measurements

Assume that the evolution of the position of an object is modeled as a polynomial in time, that is,

$$\xi(t) = \sum_{j=0}^n a_j \frac{t^j}{j!} \quad (3.5.2-1)$$

with the parameters being the polynomial coefficients  $a_j$ ,  $j = 0, 1, \dots, n$ , to be estimated.

The coefficient  $a_j$  is the  $j$ th derivative of the position at the **reference time**  $t = 0$ .

The LS technique from Section 3.4.1 will be used to estimate these parameters via **polynomial fitting** — of order  $n$  in this case.

The noisy measurements of the position (3.5.2-1) can be written as

$$z(i) = h(i)'a + w(i) \quad i = 1, \dots, k \quad (3.5.2-2)$$

where

$$a = [a_0 \ a_1 \ \cdots \ a_n]' \quad (3.5.2-3)$$

is the  $(n+1)$ -dimensional parameter vector to be estimated and the row vector

$$h(i)' = \left[ 1 \ t_i \ \cdots \ \frac{t_i^n}{n!} \right] \quad (3.5.2-4)$$

plays the role of  $H(i)$  from (3.4.1-1).

The measurement disturbances  $w(i)$  are assumed to be a zero-mean white sequence with a known variance  $\sigma^2$ .

The stacked measurement matrix (3.4.1-5) is

$$H^k = \begin{bmatrix} h(1)' \\ \vdots \\ h(k)' \end{bmatrix} \quad (3.5.2-5)$$

Then, since

$$R^k = \sigma^2 I \quad (3.5.2-6)$$

one has

$$\begin{aligned} H^{k'}(R^k)^{-1}H^k &= \sigma^{-2} H^{k'} H^k = \sigma^{-2} [h(1) \ \cdots \ h(k)] \begin{bmatrix} h(1)' \\ \vdots \\ h(k)' \end{bmatrix} \\ &= \sigma^{-2} \sum_{i=1}^k h(i)h(i)' \end{aligned} \quad (3.5.2-7)$$

The estimate of the parameter vector  $a$  is then, using (3.4.1-9),

$$\hat{a}(k) = \left\{ \sum_{i=1}^k h(i)h(i)' \right\}^{-1} \sum_{i=1}^k h(i)z(i) \quad (3.5.2-8)$$

with the covariance matrix

$$P(k) = \sigma^2 \left\{ \sum_{i=1}^k h(i)h(i)' \right\}^{-1} \quad (3.5.2-9)$$

Note that because the term to be inverted in (3.5.2-8) and in (3.5.2-9) is an  $(n+1) \times (n+1)$  matrix that is the sum of  $k$  dyads, one needs  $k \geq n+1$  in order for the inverse to exist; that is, *at least as many measurements as the number of the parameters are needed.*

Using (3.5.2-4) one can write the  $i$ th dyad in (3.5.2-8) or (3.5.2-9) as

$$h(i)h(i)' = \begin{bmatrix} 1 & t_i & \cdots & t_i^n/n! \\ t_i & t_i^2 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ t_i^n/n! & \cdots & (t_i^n/n!)^2 \end{bmatrix} \quad (3.5.2-10)$$

With this, explicit expressions can be obtained for (3.5.2-8) and (3.5.2-9) if the samples are uniformly spaced.

Let

$$t_i = \frac{2i - k - 1}{2}T \quad i = 1, \dots, k \quad (3.5.2-11)$$

where  $T$  is the sampling period and the sampling times are centered around  $t = 0$  for convenience.

Then the parameter  $a_j$  is the  $j$ th derivative of the position *at the center of the batch*, which is the **reference time**.

The mapping of these estimates to an *arbitrary time* is presented in Subsection 3.5.3.

The explicit expressions of the parameter estimates and their covariances for polynomials of order  $n = 1, 2, 3$  are given next.

### First-Order Polynomial

For  $n = 1$ , which corresponds to a **constant velocity motion (straight-line fit)**, one has

$$P(k) = \frac{\sigma^2}{k} \begin{bmatrix} 1 & 0 \\ 0 & \frac{12}{(k-1)(k+1)T^2} \end{bmatrix} \quad (3.5.2-12)$$

and

$$\begin{bmatrix} \hat{a}_0(k) \\ \hat{a}_1(k) \end{bmatrix} = \sigma^{-2} P(k) \begin{bmatrix} \sum_{i=1}^k z(i) \\ \sum_{i=1}^k z(i)t_i \end{bmatrix} \quad (3.5.2-13)$$

In the above  $\hat{a}_0$  and  $\hat{a}_1$  are the position and velocity estimates, respectively, at the *center of the batch*, which corresponds to the reference time  $t = 0$ .

Note that (3.5.2-12) and (3.5.2-13) are equivalent to the results of the example from Subsection 3.5.1. The only difference is that in the latter the parameters were the initial position and velocity.

## Second-Order Polynomial

For  $n = 2$ , which corresponds to a **constant acceleration motion (parabolic fit)**, one has

$$P(k) = \sigma^2 \begin{bmatrix} \frac{3(3k^2-7)}{4k(k^2-4)} & 0 & \frac{-30}{k(k^2-4)T^2} \\ 0 & \frac{12}{k(k^2-1)T^2} & 0 \\ \frac{-30}{k(k^2-4)T^2} & 0 & \frac{720}{k(k^2-1)(k^2-4)T^4} \end{bmatrix} \quad (3.5.2-14)$$

and

$$\hat{a}(k) = \begin{bmatrix} \hat{a}_0(k) \\ \hat{a}_1(k) \\ \hat{a}_2(k) \end{bmatrix} = \sigma^{-2} P(k) \begin{bmatrix} \sum_{i=1}^k z(i) \\ \sum_{i=1}^k z(i)t_i \\ \sum_{i=1}^k z(i)t_i^2/2 \end{bmatrix} \quad (3.5.2-15)$$

In the above  $\hat{a}_0$ ,  $\hat{a}_1$ , and  $\hat{a}_2$  are the position, velocity, and acceleration estimates, respectively, for the center of the batch ( $t = 0$ ).

## Third-Order Polynomial

For  $n = 3$ , which corresponds to a **constant jerk motion (cubic fit)**, one has the following parameter estimate covariance:

$$P(k) = \sigma^2 \begin{bmatrix} \frac{3(3k^2-7)}{4k(k^2-4)} & 0 & \frac{-30}{k(k^2-4)T^2} & 0 \\ 0 & \frac{25(3k^4-18k^2+31)}{k(k^2-1)(k^2-4)(k^2-9)T^2} & 0 & \frac{-840(3k^2-7)}{k(k^2-1)(k^2-4)(k^2-9)T^4} \\ \frac{-30}{k(k^2-4)T^2} & 0 & \frac{720}{k(k^2-1)(k^2-4)T^4} & 0 \\ 0 & \frac{-840(3k^2-7)}{k(k^2-1)(k^2-4)(k^2-9)T^4} & 0 & \frac{100,800}{k(k^2-1)(k^2-4)(k^2-9)T^6} \end{bmatrix} \quad (3.5.2-16)$$

The parameter estimates are given by

$$\hat{a}(k) = \begin{bmatrix} \hat{a}_0(k) \\ \hat{a}_1(k) \\ \hat{a}_2(k) \\ \hat{a}_3(k) \end{bmatrix} = \sigma^{-2} P(k) \begin{bmatrix} \sum_{i=1}^k z(i) \\ \sum_{i=1}^k z(i)t_i \\ \sum_{i=1}^k z(i)t_i^2/2 \\ \sum_{i=1}^k z(i)t_i^3/6 \end{bmatrix} \quad (3.5.2-17)$$

## Remark

Comparing (3.5.2-12), (3.5.2-14) and (3.5.2-16), it can be seen that as the order of the polynomial fit increases, the parameter variances increase. This is because there is *less information per parameter* when more parameters are fitted to the same number of data points.

This increase takes place (for the present choice of the parameters — position and its derivatives at the center of the batch) for every other parameter. For example, comparing (3.5.2-14) with (3.5.2-16) indicates that

$$P_{11}(k, n = 3) = P_{11}(k, n = 2) \quad (3.5.2-18)$$

$$P_{22}(k, n=3) > P_{22}(k, n=2) \quad (3.5.2-19)$$

A second argument, indicating the order of the polynomial fit, has been used in the above equations to distinguish between the various covariance matrices.

### 3.5.3 Mapping of the Estimates to an Arbitrary Time

Assume a constant acceleration motion characterized by the 3-dimensional vector consisting of position, velocity, and acceleration at the reference time (center of the batch) estimated in (3.5.2-15). The estimate of the corresponding position-velocity-acceleration vector  $x(t)$  at an arbitrary time  $t$ , that is, the *prediction* (or *extrapolation*) based on  $k$  measurements uniformly spaced according to (3.5.2-11), is

$$\hat{x}(t|k) = \Phi(t)\hat{a}(k) \quad (3.5.3-1)$$

where  $\hat{a}(k)$  is given in (3.5.2-15), and

$$\Phi(t) = \begin{bmatrix} 1 & t & t^2/2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix} \quad (3.5.3-2)$$

The corresponding covariance is

$$P(t|k) = \Phi(t)P(k)\Phi(t)' \quad (3.5.3-3)$$

where  $P(k)$  is given in (3.5.2-14).

Similar transformations are used for lower- or higher-dimensional parameter vectors.

As an example, for the 2-dimensional case (straight line fitting, i.e., constant velocity assumption) it can be shown that the closed-form expression of the covariance matrix for the *one-step prediction* ( $t = t_{k+1} = (k + 1)T/2$ ) is

$$P(t_{k+1}|k) \stackrel{\Delta}{=} P(k+1|k) = \frac{\sigma^2}{(k-1)k} \begin{bmatrix} 2(2k+1) & 6/T \\ 6/T & \frac{12}{(k+1)T^2} \end{bmatrix} \quad (3.5.3-4)$$

Based on (3.5.3-4), Table 3.5.3-1 shows the values of the normalized one-step position prediction variances for a linear extrapolation based on  $k$  uniformly spaced observations and the corresponding velocity estimate variances.

In tracking, when the motion of the object of interest is, say, with constant velocity, and one has  $k$  measurements, a *prediction of the location* of the next measurement  $z(k+1)$  at  $t = t_{k+1} = (k + 1)T/2$  is made according to the expression

$$\hat{x}_1(t_{k+1}|k) \stackrel{\Delta}{=} \hat{x}_1(k+1|k) = \hat{a}_1 + \hat{a}_2 t_{k+1} \quad (3.5.3-5)$$

**Table 3.5.3-1:** One-step prediction variances for linear extrapolation from  $k$  measurements.

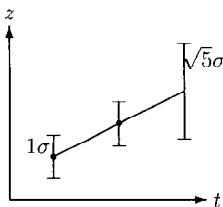
$k$	2	3	4	5	6
$P_{11}(k+1 k)/\sigma^2$	5	2.33	1.5	1.1	0.867
$T^2 P_{22}(k+1 k)/\sigma^2$	2	0.5	0.2	0.1	0.057

The measurement  $z(k+1)$  will be, with a certain probability, in a *region around this predicted location*, called **gate**, determined by the variance associated with the prediction,  $P_{11}(k+1|k)$ , given above.

In practice one can use a “ $3\sigma$  gate.” Then the measurement will be in this gate with a probability of 99.8% under the Gaussian assumption (see table 1.5.4-1). Such a gate is the interval

$$\left[ \hat{x}_1(k+1|k) - 3\sqrt{P_{11}(k+1|k)}, \hat{x}_1(k+1|k) + 3\sqrt{P_{11}(k+1|k)} \right] \quad (3.5.3-6)$$

Figure 3.5.3-1 illustrates the **one sigma region** corresponding to the one-step position prediction based on two observations.

**Figure 3.5.3-1:** Uncertainty for a one-step position prediction (straight-line motion).

Similarly, the position variance for an  $n$ -step prediction via linear extrapolation is

$$P_{11}(k+n|k) = 2\sigma^2 \frac{(k-1)(2k-1) + 6n(k-1) + 6n^2}{(k-1)(k+1)k} \quad (3.5.3-7)$$

For example,

$$P_{11}(4|2) = 13\sigma^2 \quad P_{11}(5|3) = \frac{29}{6}\sigma^2 \quad (3.5.3-8)$$

These values are useful in **track continuation** when the motion of the target is assumed to be described by a straight line. For example, if the target was not detected at  $t_{k+1}$ , then one has to wait until  $t_{k+2}$ . Note the increase of the two-step prediction variances compared to the one-step prediction variances from Table 3.5.3-1.

### 3.5.4 Polynomial Fitting — Summary

Polynomials in time can be fitted to a set of arbitrarily spaced data points (noisy position measurements) via the least squares method.

Explicit expressions have been presented for the coefficients of polynomials up to third order fitted to a set of uniformly spaced data points.

The coefficients, as presented, yield estimates of the position and its derivatives at the center of the batch of measurements (the reference time).

The covariances associated with estimates also have explicit expressions.

A simple linear mapping transforms these coefficients to the corresponding estimates to an arbitrary time — this yields the extrapolation or prediction of the motion. The same transformation matrix can be used to obtain the covariance of the position and its derivatives at an arbitrary time.

The predicted position can be used as the center of the region in which the next measurement will be with a high probability, called a *gate* or a *spatial window*. The size of the gate is determined from the variance of this prediction.

## 3.6 GOODNESS-OF-FIT AND STATISTICAL SIGNIFICANCE OF PARAMETER ESTIMATES

### 3.6.1 Hypothesis Testing Formulation of the Problem

When a set of parameters is estimated to fit a polynomial to a number of data points, there is always the following question:

What is the appropriate order of the polynomial?

The following fundamental result of estimation/polynomial fitting is relevant to this question:

*Theorem.* Through *any three points* on a sheet of paper, one can pass a straight line.<sup>3</sup>

Given a set of data points (scalar measurements), when fitting a polynomial to these points, one can encounter the following situations:

- If the order of the polynomial is too low — that is, *underfitting* — then the fit will be poor,

or, at the other extreme,

- If the order of the polynomial is too high — that is, *overfitting* — the estimates of some of the parameters are not “statistically significant,” that is, they are “noise.”

---

<sup>3</sup>Proof: Left as a trivial exercise to the reader. (*Hint:* Use a pencil that is thick enough.)

As shown in the next subsection, the sum of the squares of the residuals in an LS estimation problem — the minimized value of the LS criterion (3.4.1-2) — also called the **goodness-of-fit** or **f fitting error**,

$$J^*(k) = [z^k - H^k \hat{x}(k)]' (R^k)^{-1} [z^k - H^k \hat{x}(k)] \quad (3.6.1-1)$$

has, if the noises are Gaussian, a chi-square distribution with  $kn_z - n_x$  degrees of freedom. In the above, the notations (3.4.1-4) to (3.4.1-7) and (3.4.1-9) have been used and  $k$  is the number of measurements of dimension  $n_z$ , while  $n_x$  is the dimension of the parameter vector.

The matrix  $R^k$ , consisting of the noise covariances, is assumed to be known — a similar result is available for the situation where this matrix is unknown and estimated together with the “regression coefficients”  $x$ .

### Test for Underfitting

The order of the polynomial fit to a set of data points is *too low* if the fit is not “good enough,” that is,

$$J^* > c = \chi_{kn_z - n_x}^2(1 - \alpha) \quad (3.6.1-2)$$

where  $c$  is obtained from Table 1.5.4-1 such that the probability of a  $kn_z - n_x$  degrees of freedom chi-square random variable exceeding it is  $\alpha$  (usually 5% or 1%).

If the first choice of the polynomial is too low, one can increase it until an “acceptable” fit is obtained, that is, the resulting  $J^*$  falls below the maximum allowed.

### Test for Overfitting

If the order of the polynomial is *too high*, then the estimate (usually of the highest power coefficient) will be **statistically insignificant**.

Assuming the noises to be normal with zero mean and known variances, the estimate of the  $i$ -th component of the parameter vector is

$$\hat{x}_i(k) \sim \mathcal{N}[x_i, P_{ii}(k)] \quad (3.6.1-3)$$

that is, it is normal with mean equal to the unknown true value  $x_i$  and with variance  $P_{ii}(k)$ .

The **parameter estimate significance test** is the test between the following hypotheses:

$$H_0 : x_i = 0 \quad (3.6.1-4)$$

$$H_1 : x_i \neq 0 \quad (3.6.1-5)$$

subject to

$$P\{\text{accept } H_1 | H_0 \text{ true}\} = \alpha \quad (3.6.1-6)$$

Then one accepts  $H_1$  (i.e., that the parameter is nonzero) if and only if

$$\frac{|\hat{x}_i(k)|}{[P_{ii}(k)]^{1/2}} > c' = \mathcal{G} \left( 1 - \frac{\alpha}{2} \right) \quad (3.6.1-7)$$

The above implies a *two-sided probability region*. The threshold  $c'$  is obtained from the tables of Subsection 1.5.4 such that the probability of a standard Gaussian random variable exceeding it is  $\alpha/2$ . For example, for  $\alpha = 5\%$ , one has  $c' = 1.96$ .

If (3.6.1-7) does not hold, then the estimate of the parameter is *statistically insignificant* and it is better to accept  $H_0$ . Then the problem is solved again for a lower-dimension parameter obtained by deleting the component found insignificant.

### 3.6.2 The Fitting Error in a Least Squares Estimation Problem

Consider the LS estimation of the  $n_x$ -dimensional vector  $x$  based on  $k$  measurements  $z(i)$  of dimension  $n_z$  described in (3.4.1-1). The stacked vector of measurements, of dimension  $kn_z$ , is

$$z^k = H^k x + w^k \quad (3.6.2-1)$$

and the estimate was obtained as (the superscripts will be dropped for simplicity in the sequel)

$$\hat{x} = (H'R^{-1}H)^{-1}H'R^{-1}z \quad (3.6.2-2)$$

We want to evaluate the minimized value of the criterion (3.4.1-2), that is, the *fitting error*, or the *goodness-of-fit*, or the *norm of the residual*

$$J^* \triangleq (z - H\hat{x})' R^{-1} (z - H\hat{x}) \quad (3.6.2-3)$$

Note that the above, which is the sum of the squares of the normalized residuals, is a scalar and a (physically) *dimensionless quantity*.

The vector residual is

$$\begin{aligned} z - H\hat{x} &= Hx + w - H(H'R^{-1}H)^{-1}H'R^{-1}(Hx + w) \\ &= [I - H(H'R^{-1}H)^{-1}H'R^{-1}]w \end{aligned} \quad (3.6.2-4)$$

where  $I$  in (3.6.2-4) denotes the  $kn_z \times kn_z$  identity matrix. This follows from the fact that the stacked vectors  $z$  and  $w$  are of dimension  $kn_z$ .

Using (3.6.2-4) in (3.6.2-3) yields

$$\begin{aligned} J^* &= w'[I - H(H'R^{-1}H)^{-1}H'R^{-1}]'R^{-1}[I - H(H'R^{-1}H)^{-1}H'R^{-1}]w \\ &= w'[R^{-1} - R^{-1}H(H'R^{-1}H)^{-1}H'R^{-1}]w \\ &= w'R^{-1/2}[I - R^{-1/2}H(H'R^{-1}H)^{-1}H'R^{-1/2}]R^{-1/2}w \\ &\triangleq \omega'A\omega \end{aligned} \quad (3.6.2-5)$$

where  $R^{-1/2}$  denotes a *square root* of  $R^{-1}$  (see (1.3.2-15)),

$$\omega \triangleq R^{-1/2}w \quad (3.6.2-6)$$

is a vector of dimension  $kn_z$ , and

$$A \triangleq I - R^{-1/2}H(H'R^{-1}H)^{-1}H'R^{-1/2} \quad (3.6.2-7)$$

Assuming

$$w \sim \mathcal{N}(0, R) \quad (3.6.2-8)$$

it follows that

$$\omega \sim \mathcal{N}(0, I) \quad (3.6.2-9)$$

that is, the components of  $\omega$  are *independent standardized Gaussians*.

It can be easily verified that the symmetric matrix  $A$  defined in (3.6.2-7) is *idempotent* (see (1.3.2-14)), that is,

$$AA = A \quad (3.6.2-10)$$

Such a matrix can have eigenvalues equal to 0 or 1 only. Furthermore,

$$\begin{aligned} \text{tr}(A) &= \text{tr}[I - R^{-1/2}H(H'R^{-1}H)^{-1}H'R^{-1/2}] \\ &= \text{tr}(I) - \text{tr}[R^{-1/2}H(H'R^{-1}H)^{-1}H'R^{-1/2}] \end{aligned} \quad (3.6.2-11)$$

Note that, since the dimension of  $I$  is  $kn_z \times kn_z$ , one has

$$\text{tr}(I) = kn_z \quad (3.6.2-12)$$

Using circular permutations for matrices multiplying each other under the trace operator (1.3.2-13) yields

$$\begin{aligned} \text{tr}[R^{-1/2}H(H'R^{-1}H)^{-1}H'R^{-1/2}] &= \text{tr}[(H'R^{-1}H)^{-1}H'R^{-1/2}R^{-1/2}H] \\ &= \text{tr}[I_{n_x}] = n_x \end{aligned} \quad (3.6.2-13)$$

where  $I_{n_x}$  is the  $n_x \times n_x$  identity matrix. This follows from the fact that the dimension of  $H'R^{-1}H$  is  $n_x \times n_x$ .

Thus, combining (3.6.2-12) and (3.6.2-13) into (3.6.2-11) results in

$$\text{tr}(A) = kn_z - n_x \quad (3.6.2-14)$$

that is, the matrix  $A$  in (3.6.2-5), which has dimension  $kn_z \times kn_z$ , has  $kn_z - n_x$  unity eigenvalues and  $n_x$  zero eigenvalues.

Now, since  $\omega$ , defined in (3.6.2-6), is zero mean normal with identity covariance matrix (of dimension  $kn_z \times kn_z$ ) and  $A$  has  $kn_z - n_x$  unity eigenvalues (and the rest zero), the *fitting error*

$$J^* = \omega' A \omega \quad (3.6.2-15)$$

is, as shown next, the sum of the squares of  $kn_z - n_x$  independent scalar random variables that are normal with mean zero and unity variance.

**Proof**

The spectral representation (1.3.6-12) of the symmetric matrix  $A$  is

$$A = \sum_{i=1}^{kn_z} \lambda_i u_i u_i' \quad (3.6.2-16)$$

where  $\lambda_i$  are the eigenvalues of  $A$ , and  $u_i$  are its normalized eigenvectors that are orthogonal to each other, that is,

$$u_i' u_j = \delta_{ij} \quad (3.6.2-17)$$

Such vectors are called **orthonormal**.

Using (3.6.2-16) in (3.6.2-5), one has

$$\begin{aligned} \omega' A \omega &= \omega' \sum_{i=1}^{kn_z} \lambda_i u_i u_i' \omega = \sum_{i=1}^{kn_z} \lambda_i \omega' u_i u_i' \omega \\ &= \sum_{i=1}^{kn_z} \lambda_i u_i' \omega u_i' \omega = \sum_{i=1}^{kn_z} \lambda_i (u_i' \omega)^2 \\ &\triangleq \sum_{i=1}^{kn_z} \lambda_i \xi_i^2 \end{aligned} \quad (3.6.2-18)$$

where

$$\xi \triangleq \text{col}(\xi_i) \sim \mathcal{N}(0, I) \quad (3.6.2-19)$$

since  $\xi_i$  is a linear combination of the components of  $\omega$  and

$$\begin{aligned} E[\xi_i \xi_j] &= E[\omega' u_i \omega' u_j] = E[u_i' \omega \omega' u_j] \\ &= u_i' E[\omega \omega'] u_j = u_i' I u_j = u_i' u_j = \delta_{ij} \end{aligned} \quad (3.6.2-20)$$

that is,  $\xi_i$ ,  $i = 1, \dots, kn_z$ , are **independent standardized Gaussians**.

Therefore, since in (3.6.2-18) there are  $kn_z - n_x$  unity eigenvalues  $\lambda_i$  and the rest are zero, it follows that it is the sum of  $kn_z - n_x$  independent standardized Gaussians squared; that is, it has a chi-square distribution with  $kn_z - n_x$  degrees of freedom. Therefore,

$$\omega' A \omega = \sum_{i=1}^{kn_z} \lambda_i \xi_i^2 = \sum_{i=1}^{kn_z - n_x} \xi_i^2 \quad (3.6.2-21)$$

and, consequently, from (3.6.2-5)

$J^* \sim \chi^2_{kn_z - n_x}$

(3.6.2-22)

**Remark**

The meaning of **degrees of freedom** can be seen as being the number of (scalar) observations,  $kn_z$ , minus the number of parameters estimated,  $n_x$ .

### 3.6.3 A Polynomial Fitting Example

Consider the following numerical example. The true parameter vector is assumed to be of dimension 3; that is, a constant acceleration motion (polynomial of order 2) models the truth.

The measurements are given as in (3.5.2-2) by

$$z(i) = x_1 + x_2 t_i + x_3 \frac{t_i^2}{2} + w(i) \quad (3.6.3-1)$$

where

$$x_1 = 10 \quad (3.6.3-2)$$

$$x_2 = 1 \quad (3.6.3-3)$$

$$x_3 = 0.2 \quad (3.6.3-4)$$

$$t_i = \frac{1}{2}(2i - k - 1) \quad i = 1, \dots, k; \quad k = 15 \quad (3.6.3-5)$$

with the noise sequence white and

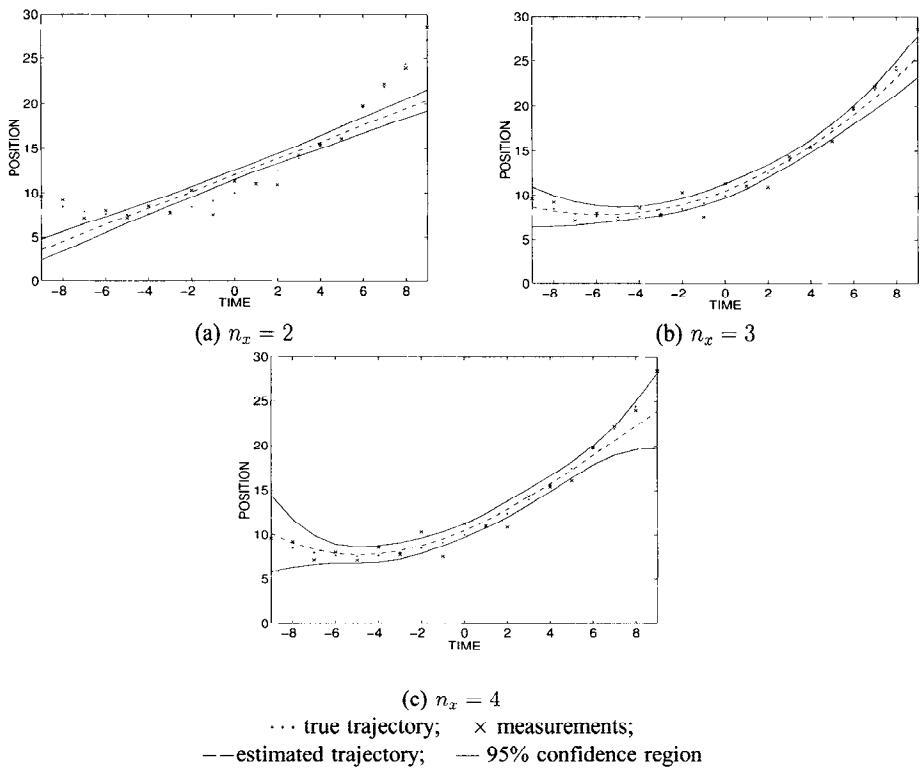
$$w(i) \sim \mathcal{N}(0, 1) \quad (3.6.3-6)$$

Table 3.6.3-1 presents the results of the fitting of a sequence of measurements generated according to the above model with polynomials of order 1, 2, and 3.

**Table 3.6.3-1:** Fitting of various-order polynomial models.

Assumed model Order $n_x$	Linear 2	Quadratic 3	Cubic 4
$\hat{x}_1$	11.97	10.34	10.34
$\sqrt{P_{11}}$	0.258	0.388	0.388
$ \hat{x}_1 /\sqrt{P_{11}}$	46.38	26.60	26.60
$\hat{x}_2$	0.996	0.997	1.114
$\sqrt{P_{22}}$	0.059	0.059	0.151
$ \hat{x}_2 /\sqrt{P_{22}}$	16.67	16.67	7.341
$\hat{x}_3$		0.174	0.174
$\sqrt{P_{33}}$		0.031	0.031
$ \hat{x}_3 /\sqrt{P_{33}}$		5.611	5.611
$\hat{x}_4$			0.021
$\sqrt{P_{44}}$			0.025
$ \hat{x}_4 /\sqrt{P_{44}}$			0.843
$J^*$	44.67	13.18	12.47
$\chi^2_{k-n_x}(95\%)$	22.4	21.0	19.7

For the linear model ( $n_x = 2$ ) the fitting error  $J^*$  is too large — well above the threshold, chosen as 95% point from the chi-square tables.



**Figure 3.6.3-1:** Fitting of various-order polynomials to a constant acceleration motion.

For the quadratic model ( $n_x = 3$ ),  $J^*$  is below the threshold (i.e., acceptable) and all the parameter estimates are significant.

For the cubic model ( $n_x = 4$ ) the last parameter (third derivative of the position — jerk) estimate is statistically insignificant:  $0.843 < \mathcal{G}(97.5\%) = 1.96$ ; the fit has only slightly improved (the so-called F test would indicate that this improvement is statistically insignificant — this is equivalent to the parameter estimate significance test).

Thus  $n_x = 4$  is clearly an “overparametrization.” It can also be seen that for  $n_x = 4$  the velocity standard deviation  $\sqrt{P_{22}}$  is *much larger* than for  $n_x = 3$  because of the additional parameter to be estimated.

Figure 3.6.3-1 illustrates the resulting fit with these three polynomials. Also shown are the “uncertainty tubes” — the  $2\sigma$  confidence region (95%) around the predicted position. Note how much more rapidly this widens for the third-order polynomial fit ( $n_x = 4$ ) in comparison with the second-order ( $n_x = 3$ ) case.

### 3.6.4 Order Selection in Polynomial Fitting — Summary

*Fitting of polynomials* to noisy measurements, which is a particular case of regression, consists of LS estimation of its coefficients.

The *order* of the polynomial chosen for fitting is

1. *too low* if the *fitting error* is too high (sum of squares of the residuals is too large);
2. *too high* if some estimates of coefficients are *statistically insignificant* (“buried in noise”).

The fitting error is, under the Gaussian assumption, chi-square distributed with number of degrees of freedom equal to the number of measurements minus the number of estimated parameters (this is the origin of the term *degrees of freedom*). The fitting error (“goodness-of-fit”) has to be *below a threshold in order to be acceptable*.

The statistical significance of a parameter estimate is measured by the ratio of the *magnitude of the parameter estimate* to its *standard deviation*. This ratio has to be *above a threshold* for the estimate to be significant.

Fitting a polynomial (or, in general, a model) of *unnecessarily high order* decreases the accuracy of some of the estimated coefficients (model parameters) — this *wastes information*.

## 3.7 USE OF LS FOR A NONLINEAR PROBLEM: BEARINGS-ONLY TARGET MOTION ANALYSIS

### 3.7.1 The Problem

The *nonlinear least squares* problem, defined in Subsection 2.4.1, will be used to estimate the motion parameters of a constant velocity target — *target motion analysis* — based on observations from a passive sensor that measures only the direction of arrival of a signal emitted by the target. The emitted energy can be acoustic or electromagnetic. This problem is also called *passive localization* or *passive ranging*.

The target, which is moving in a plane, is observed from a platform with a known position  $[\xi_p(k), \eta_p(k)]$  in the same plane. The target “localization parameter” is the vector of dimension  $n_x = 4$  consisting of its initial position and velocity in Cartesian coordinates

$$x \triangleq [x_1 \ x_2 \ x_3 \ x_4]' \triangleq [\xi(0) \ \eta(0) \ \dot{\xi} \ \dot{\eta}]' \quad (3.7.1-1)$$

The position of the target at time  $t_k$  is

$$\xi(k) \triangleq \xi(t_k) = \xi(0) + \dot{\xi}t_k = x_1 + x_3 t_k \quad (3.7.1-2)$$

$$\eta(k) \triangleq \eta(t_k) = \eta(0) + \dot{\eta}t_k = x_2 + x_4 t_k \quad (3.7.1-3)$$

It is assumed that the available measurements are bearings (“line of sight” angles with respect to some reference direction) only, given by

$$z(k) \triangleq z(t_k) = h(k, x) + w(k) \quad k = 1, \dots, n \quad (3.7.1-4)$$

where

$$h(k, x) \triangleq \tan^{-1} \frac{\eta(k) - \eta_p(k)}{\xi(k) - \xi_p(k)} \quad (3.7.1-5)$$

and  $w(k)$  is the measurement noise, assumed to be a zero-mean Gaussian white sequence with known variance  $r$ , that is,

$$E[w(k)w(j)] = r\delta_{kj} \quad (3.7.1-6)$$

### 3.7.2 Observability of the Target Parameter in Passive Localization

A question of interest is the effect of the motion of the platform on the ability to estimate the target parameters. It will be shown that if the platform moves with a constant velocity, then the target motion parameter cannot be estimated — it is *unobservable*.

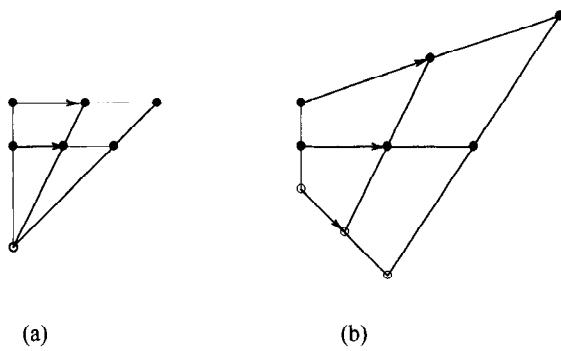
If we denote the platform velocity components as  $\dot{\xi}_p$  and  $\dot{\eta}_p$ , then the true bearing to the target at time  $t$  is

$$\begin{aligned} \tan^{-1} \frac{\eta(t) - \eta_p(t)}{\xi(t) - \xi_p(t)} &= \tan^{-1} \frac{\eta(0) + \dot{\eta}t - \eta_p(0) - \dot{\eta}_p t}{\xi(0) + \dot{\xi}t - \xi_p(0) - \dot{\xi}_p t} \\ &= \tan^{-1} \frac{\eta(0) - \eta_p(0) + (\dot{\eta} - \dot{\eta}_p)t}{\xi(0) - \xi_p(0) + (\dot{\xi} - \dot{\xi}_p)t} \\ &= \tan^{-1} \frac{[\eta(0) - \eta_p(0)]\alpha + (\dot{\eta} - \dot{\eta}_p)\alpha t}{[\xi(0) - \xi_p(0)]\alpha + (\dot{\xi} - \dot{\xi}_p)\alpha t} \quad \forall \alpha \neq 0 \end{aligned} \quad (3.7.2-1)$$

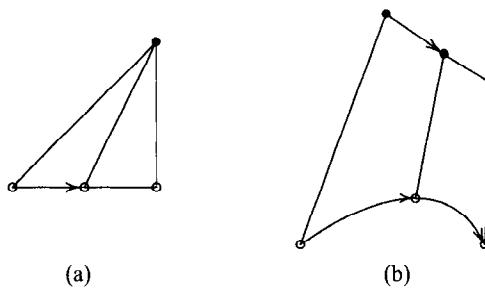
The above is seen to hold for all  $\alpha$ , that is, one will obtain the same sequence of true bearings if the relative position and relative velocity are multiplied by an arbitrary constant  $\alpha$ .

Thus, an infinite number of values of the target parameter vector can yield the same observations and, consequently, its parameter vector cannot be estimated in this case — the observability requirement of full and unique recovery of the initial state is not satisfied.

Therefore, in order to estimate the target parameter vector, the platform has to have at least an acceleration. A constant speed platform with a change of course (heading) satisfies this requirement. In general, the platform trajectory has to have *one more nonzero derivative than the target trajectory*.



**Figure 3.7.2-1:** Unobservable target (● target; ○ platform).



**Figure 3.7.2-2:** Observable target (● target; ○ platform).

Figure 3.7.2-1 illustrates two cases where the target localization parameter is not observable from angle-only measurements: (a) moving target and fixed sensor platform and (b) constant velocity target and constant velocity platform.

Figure 3.7.2-2 illustrates two observable cases: (a) fixed target and moving platform and (b) constant velocity target and platform with acceleration.

### 3.7.3 The Likelihood Function for Target Parameter Estimation

The likelihood function of the target parameter vector (3.7.1-1) is

$$\Lambda(x) = p(Z^n|x) = p[z(1), \dots, z(n)|x] = \prod_{k=1}^n p[z(k)|x] \quad (3.7.3-1)$$

where, in view of (3.7.1-4),

$$p[z(k)|x] = \mathcal{N}[z(k); h(k, x), r] = ce^{-\frac{1}{2r}[z(k) - h(k, x)]^2} \quad (3.7.3-2)$$

and  $h(k, x)$ , given in (3.7.1-5), is the expected value (average) of the observation  $z(k)$  for a given target parameter vector  $x$ , and  $r$  is its variance. This follows from the assumption (3.7.1-6) that the measurement noises are white, zero mean, and with variance  $r$ .

In view of (3.7.3-2), the maximization of (3.7.3-1), which yields the maximum likelihood estimate (MLE), is equivalent to the following **nonlinear least squares (NLS)** problem:

$$\hat{x} = \arg \max_x \Lambda(x) = \arg \min_x \lambda(x) \quad (3.7.3-3)$$

where

$$\lambda(x) \triangleq \frac{1}{2r} \sum_{k=1}^n [z(k) - h(k, x)]^2 \quad (3.7.3-4)$$

is the **negative log-likelihood function** with the irrelevant additive constants omitted. The above expression of the log-likelihood function is clearly that of an NLS criterion.

The minimization of the log-likelihood function (3.7.3-4) can be carried out via one of the many existing numerical optimization algorithms. The Newton-Raphson or quasi-Newton techniques are the most effective in this case.

### 3.7.4 The Fisher Information Matrix for the Target Parameter

The **Cramer-Rao lower bound (CRLB)** on the covariance matrix of the target parameter estimate  $\hat{x}$  is (assuming this estimate to be unbiased)

$$E[(\hat{x} - x)(\hat{x} - x)'] \geq J^{-1} \quad (3.7.4-1)$$

where  $J$  is the **Fisher information matrix (FIM)**

$$\begin{aligned} J &= E\{[\nabla_x \ln \Lambda(x)][\nabla_x \ln \Lambda(x)]'\} \Big|_{x=x_0} \\ &= E\{[\nabla_x \lambda(x)][\nabla_x \lambda(x)]'\} \Big|_{x=x_0} \end{aligned} \quad (3.7.4-2)$$

The FIM is to be evaluated at the true value of the parameter  $x_0$ ; in practice, when this is not known the evaluation is done at the estimate.

The gradient of the log-likelihood function (3.7.3-4) is

$$\nabla_x \lambda(x) = -r^{-1} \sum_{k=1}^n [\nabla_x h(k, x)][z(k) - h(k, x)] \quad (3.7.4-3)$$

which, when inserted into (3.7.4-2), yields (see problem 3-4) the sum of dyads

$$J = r^{-1} \sum_{k=1}^n h_x(k, x)h_x(k, x)' \Big|_{x=x_0} \quad (3.7.4-4)$$

where

$$h_x(k, x) \triangleq \nabla_x h(k, x) \quad (3.7.4-5)$$

## Parameter Observability and the FIM

In order to have **parameter observability** (i.e., to allow its estimation without the ambiguity discussed in Subsection 3.7.2), the FIM (3.7.4-2) *must be invertible*. If the FIM is not invertible (i.e., it is singular), then the lower bound (3.7.4-1) will not exist (actually, it will have one or more infinite eigenvalues, which means total uncertainty in a subspace of the parameter space, i.e., ambiguity). This amounts to the information limit being insufficient for the estimation problem at hand, i.e., *insufficient information in the data*.

## The Normalized Estimation Error and Estimation Efficiency

Assume that the parameter estimation error

$$\tilde{x} \triangleq x - \hat{x} \quad (3.7.4-6)$$

is (approximately) Gaussian distributed, with covariance given by the Cramer-Rao lower bound, that is,

$$P \triangleq E[\tilde{x}\tilde{x}'] = J^{-1} \quad (3.7.4-7)$$

where  $J$  defined in (3.7.4-4) is assumed invertible — its rank is  $n_x$  (the dimension of  $x$ ). In this case the inverse, being a covariance matrix, is *positive definite*.

The **normalized estimation error squared (NEES)** for the parameter  $x$ , defined as

$$\epsilon_x \triangleq \tilde{x}' P^{-1} \tilde{x} = \tilde{x}' J \tilde{x} \quad (3.7.4-8)$$

is chi-square distributed with  $n_x$  degrees of freedom, that is,

$$\epsilon_x \sim \chi_{n_x}^2 \quad (3.7.4-9)$$

The NEES can be used in simulations to check whether the estimator is efficient, that is, *the errors “match” the covariance given by the Cramer-Rao lower bound* — this is the **efficiency check**.

Let  $g$  be such that

$$P\{\epsilon_x \leq g^2\} = 1 - Q \quad (3.7.4-10)$$

where  $Q$  is a small “tail” probability. The NEES will have to be less than  $g^2$  with a high probability  $1 - Q$  if the estimator is efficient — this is a *one-sided probability region* for  $\epsilon_x$ .

Thus, the efficiency check of the estimator (in simulations — because this is the only situation where  $\tilde{x}$  is available) consists of verifying whether (3.7.4-10) holds. This is a “single run” test — the multiple run (Monte Carlo) test is discussed later.

### Confidence Region for the Target Parameter Vector

From (3.7.4-10) one can see that, given the true parameter  $x$ , one has, if the estimator is efficient, a “maximum distance” (squared) from it to the estimated parameter — the squared norm of the error. Conversely, when  $x$  is not known and  $\hat{x}$  has been obtained, one can say that  $x$  should be within some neighborhood of  $\hat{x}$ , determined by (3.7.4-10) — the **confidence region** for the true parameter.

The confidence region for the parameter  $x$  follows from (3.7.4-8) and from (3.7.4-10) as the (inside of the) “*g-sigma*” **ellipsoid**

$$(x - \hat{x})' P^{-1} (x - \hat{x}) = g^2 \quad (3.7.4-11)$$

The above is an ellipsoid (actually a hyperellipsoid of dimension  $n_x$ ) since  $P$  is positive definite; its semiaxes are  $g$  times the square roots of the eigenvalues of  $P$ . This region is the **probability concentration ellipsoid**, obtained by cutting the tails of the multivariate Gaussian density. The left-hand side of (3.7.4-11), which is a “normalized” distance, is also known as the **Mahalanobis distance**.

### Position Estimate at an Arbitrary Time and its Confidence Region

Similarly to Subsection 3.5.3, the 2-dimensional **position estimate for an arbitrary time**  $t$  is given by the following transformation of the target parameter vector

$$\hat{x}_p(t) = \begin{bmatrix} 1 & 0 & t & 0 \\ 0 & 1 & 0 & t \end{bmatrix} \hat{x} \triangleq \Phi_p(t) \hat{x} \quad (3.7.4-12)$$

where  $\Phi_p$  follows from the constant velocity assumption for the target.

The covariance corresponding to the above is

$$P_p(t) = \Phi_p(t) P \Phi_p(t)' \quad (3.7.4-13)$$

and the **position confidence region** is determined by the ellipse

$$[x_p - \hat{x}_p(t)]' P_p^{-1}(t) [x_p - \hat{x}_p(t)] = g_p^2 \quad (3.7.4-14)$$

where  $g_p$  is chosen for the desired probability concentration or **gate probability**.

### Remarks

The Cramer-Rao lower bound quantifies in this case the **stochastic observability**, or **estimability**, which is not a binary property as in the deterministic case:

1. Lack of invertibility of  $J$  (in practice, ill-conditioning) indicates that the parameter is *unobservable*. This happens if the condition number (1.3.6-13) of  $J$  is too large.
2. If  $J$  is invertible but the position confidence region (3.7.4-14) is “large,”<sup>4</sup> one has *marginal observability*. This occurs if the gradient vectors in (3.7.4-4) are “nearly collinear.”
3. A “small” confidence region — *good observability* — is obtained if the  $n$  gradient vectors in (3.7.4-4) span “well” the  $n_x$ -dimensional space, that is, they are “far” from being collinear. The actual measure of this is the condition number of  $J$ .

### Expressions of the Gradient Vector Components

From (3.7.1-5) one has the following expressions for the components of the gradient vector entering into the FIM

$$h_{x_1}(k, x) = -\frac{\eta(k) - \eta_p(k)}{[\xi(k) - \xi_p(k)]^2 + [\eta(k) - \eta_p(k)]^2} \quad (3.7.4-15)$$

$$h_{x_2}(k, x) = \frac{\xi(k) - \xi_p(k)}{[\xi(k) - \xi_p(k)]^2 + [\eta(k) - \eta_p(k)]^2} \quad (3.7.4-16)$$

$$h_{x_3}(k, x) = t_k h_{x_1}(k, x) \quad (3.7.4-17)$$

$$h_{x_4}(k, x) = t_k h_{x_2}(k, x) \quad (3.7.4-18)$$

### 3.7.5 The Goodness-of-Fit Test

A test based on data from a *single run*, which can be used with real data is presented next. This test does not require knowledge of the true parameter.

Similarly to the linear LS, the minimized value of the log-likelihood function (3.7.3-4), multiplied by 2 for convenience, is

$$\lambda^* \triangleq \lambda(\hat{x}) = \frac{1}{r} \sum_{k=1}^n [z(k) - h(k, \hat{x})]^2 \quad (3.7.5-1)$$

This is the *normalized sum of the squares of the residuals* or the *fitting error*, and it can be used as a measure of the goodness of fit. Note that (3.7.5-1) is a physically dimensionless quantity.

In the *linear LS* case, under the *Gaussian noise assumptions*, the fitting error was shown to be chi-square distributed in Subsection 3.6.2. In the present *nonlinear LS* problem, the same result can be assumed to hold approximately. Then, with  $n$  being the number of (scalar) measurements, one has

$$\lambda^* \sim \chi_{n-n_x}^2 \quad (3.7.5-2)$$

---

<sup>4</sup>In the eye of the engineer, like beauty in the eye of the beholder.

and a suitable probability region check can be made to ascertain that the model used for the problem is valid. Namely,  $\lambda^*$  should be, with 95% probability, below the threshold  $\chi_{n-n_x}^2(0.95)$ , with the notation (1.5.4-3).

This test can also be used with the results of *Monte Carlo runs*. In this case, by summing up the fitting error from  $N$  runs with *independent random variables*, one obtains a total error that is chi-square distributed with  $N(n - n_x)$  degrees of freedom and has to be below a threshold obtained similarly to the one discussed above.

### 3.7.6 Testing for Efficiency with Monte Carlo Runs

The practical procedure to check the estimator efficiency is using *Monte Carlo simulations* as follows. Let  $\epsilon_x^i$  be the NEES (3.7.4-8) in run  $i$ ,  $i = 1, \dots, N$ , and let the sample average NEES from  $N$  independent such runs be

$$\bar{\epsilon}_x = \frac{1}{N} \sum_{i=1}^N \epsilon_x^i \quad (3.7.6-1)$$

The quantity  $N\bar{\epsilon}_x$  is chi-square distributed with  $Nn_x$  degrees of freedom. Let the  $1 - Q$  *two-sided probability region* for  $N\bar{\epsilon}_x$  be the interval  $[\epsilon'_1, \epsilon'_2]$ . Using the notation from (1.5.4-3), we obtain

$$\epsilon'_1 = \chi_{Nn_x}^2 \left( \frac{Q}{2} \right) \quad (3.7.6-2)$$

$$\epsilon'_2 = \chi_{Nn_x}^2 \left( 1 - \frac{Q}{2} \right) \quad (3.7.6-3)$$

For example, for  $n_x = 4$ ,  $N = 25$ ,  $Q = 5\%$ , one has  $\epsilon'_1 = 74$ ,  $\epsilon'_2 = 130$ . The  $1 - Q$  ( $= 95\%$ ) two-sided probability region for  $\bar{\epsilon}_x$  is

$$[\epsilon_1, \epsilon_2] = [3, 5.2] \quad (3.7.6-4)$$

where, in view of the division by  $N$  in (3.7.6-1), one has

$$\epsilon_i = \frac{\epsilon'_i}{N} \quad i = 1, 2 \quad (3.7.6-5)$$

Thus, if the estimator is *efficient*, one has to have

$$P\{\bar{\epsilon}_x \in [\epsilon_1, \epsilon_2]\} = 1 - Q \quad (3.7.6-6)$$

### Remarks

The division by  $N$  in (3.7.6-1) is a convenience: It yields results in the neighborhood of  $n_x$  regardless of  $N$ . Note that the interval (3.7.6-4) is not symmetric about the mean  $n_x = 4$ . For a single run ( $N = 1$ ) the corresponding interval is  $[0.5, 11.1]$ ; also note how much smaller is (3.7.6-4), which corresponds to  $N = 25$ . This illustrates how Monte Carlo runs *reduce the variability*.

Another option is to use a *one-sided probability region* and check only for the upper limit (the lower limit is zero).

### 3.7.7 A Localization Example

The following example, using the software *BearDAT*<sup>TM</sup>, illustrates the problem of estimating the localization parameter of a constant velocity target based on noisy bearings-only observations. The target parameter vector is

$$\boldsymbol{x} = [10,000 \text{ m} \ 20,000 \text{ m} \ 0 \text{ m/s} \ -10 \text{ m/s}]' \quad (3.7.7-1)$$

Bearing measurements with standard deviation of  $1^\circ$  are made from a sensor moving on a platform, every 4 s over a total period of 900 s. The measurement noises are independent, identically distributed Gaussian with mean zero.

Two scenarios are depicted in Fig. 3.7.7-1 — they differ in the trajectory of the sensor platform. The target moves over the period of 900 s from its initial location, designated as “ $\times$ ”. At the same time, the sensor platform moves in a two-leg constant speed trajectory starting from point “ $\circ$ ”. This sensor motion makes the target localization parameter observable.

The 95% probability mass ellipses around the target’s initial and final position based on the Cramer-Rao lower bound, also shown, quantify the stochastic observability of the target’s localization parameter. The target’s final position uncertainty ellipse has been obtained using the transformation technique discussed in Section 3.4.

Figure 3.7.7-1b illustrates the sequences of bearing measurements obtained in one realization of each scenario.

In scenario 2, the resulting target localization uncertainty is larger than in scenario 1: The change of course in the platform trajectory leads to less information from the measurements.

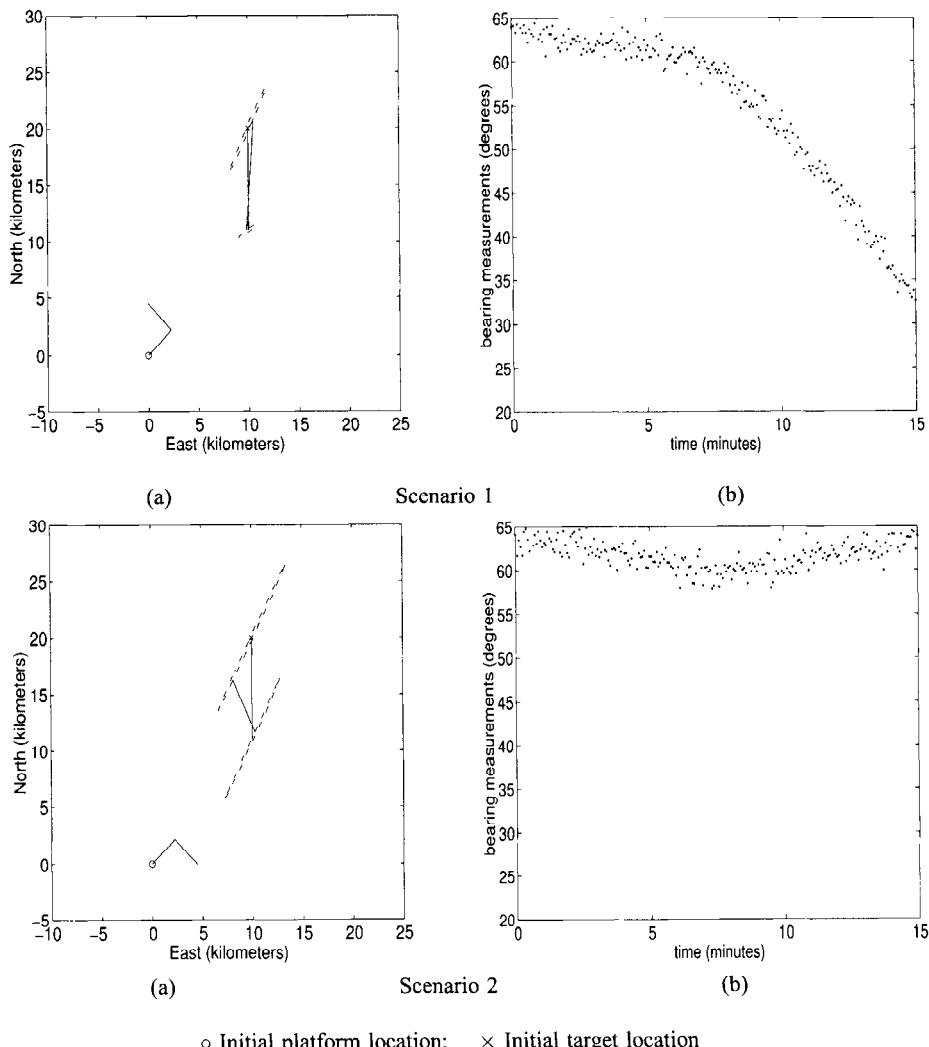
Monte Carlo simulations can be used to verify estimator efficiency, that is, the *validity of the Cramer-Rao lower bound as the actual parameter estimate’s covariance matrix*. This makes it possible to obtain a *confidence region for the location of the target* at any time (under the constant velocity motion assumption).

Figure 3.7.7-2 shows the scattering of the initial and final position estimates for 100 runs of scenario 1. As can be seen, the 95% ellipses for the position indeed contain all but three of the estimated positions.

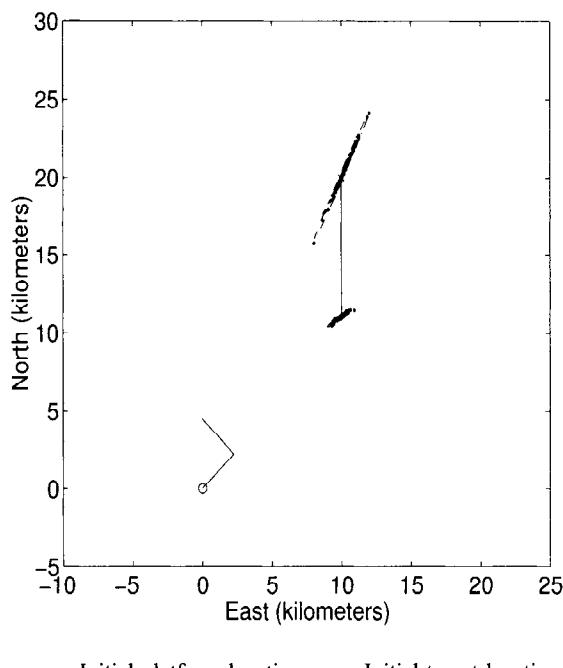
The sample average of the NEES (3.7.4-8) was obtained from these 100 runs as 3.87, which is very close to the theoretical value of  $n_x = 4$ . This confirms the validity of the *CRLB as the actual covariance* for the present problem.

### 3.7.8 Passive Localization — Summary

The technique of least squares has been illustrated for the problem of estimating the localization parameter of a constant velocity target based on noisy bearings-only observations. The noises were assumed independent and identically distributed Gaussian with zero mean and known variance.



**Figure 3.7.7-1:** (a) Target and platform trajectories with target localization uncertainty and its estimated trajectory. (b) Sequence of bearing measurements (one realization).



**Figure 3.7.7-2:** Estimated target positions in 100 runs (scenario 1).

In order for the target parameter vector to be observable, the platform, from which the measurements are made, has to undergo an acceleration. A constant-speed platform motion with a change of course satisfies this requirement.

The likelihood function of the target parameter is a nonlinear function of the parameter, and therefore a numerical search technique (Newton-Raphson or quasi-Newton) is needed to find the MLE.

The FIM for this problem has been obtained, which allows the evaluation of the Cramer-Rao lower bound for the parameter estimate's covariance.

The target's position estimate at an *arbitrary time* and the corresponding covariance matrix have been obtained.

The goodness-of-fit test can be used to ascertain the acceptability of the parameter estimate — the fitting error has to be below a certain threshold.

Monte Carlo simulations can be used to verify estimator efficiency, that is, the *validity of the CRLB as the actual parameter estimate's covariance matrix*. This makes it possible to obtain a *confidence region for the location of the target* at any time (under the constant velocity motion assumption).

## 3.8 NOTES, PROBLEMS AND A PROJECT

### 3.8.1 Bibliographical Notes

The static linear estimation material can be also found in [Sage71] and, in more detail — with simultaneous estimation of the parameters and the noise variance — in statistics or econometrics texts, for example, [Johnston72]. The reason the noise variance was assumed known here is that in engineering systems the measurements are obtained from a sensor and signal processor whose accuracy is (usually) known. In contrast, in statistics the noises reflect modeling errors and their variances are not known. The concepts of goodness of fit and statistical significance are treated mainly in statistics and econometrics texts, for example, [Johnston72]. This text (and most other statistics texts) present in detail the general regression theory where the variance is estimated together with the regression coefficients, as well as the t test and the F test used for significance testing in this case.

The software *BearDAT™* is based on [Kirubarajan96]. The general observability conditions for the passive localization of Section 3.7 are discussed in [Fogel88].

### 3.8.2 Problems

- 3-1 MMSE estimation with correlated noises.** Given the prior information  $x \sim \mathcal{N}(\bar{x}, \sigma_0^2)$  and the measurements

$$z(j) = x + w(j) \quad j = 1, 2$$

with the jointly Gaussian measurement noises  $w(j) \sim \mathcal{N}(0, \sigma^2)$  independent of  $x$  but correlated among themselves, with

$$E[w(1)w(2)] = \rho\sigma^2$$

1. Find the variance of the MMSE estimator of  $x$  conditioned on these measurements.
2. What is the “effective number” of measurements (the number of measurements with independent noises with the same variance  $\sigma^2$ , which yield the same variance  $P_{xx|z}$  for the MMSE estimator)?
3. If  $\bar{x} = 10$ ,  $\sigma_0 = 1$ , and  $\rho = 0.5$ , how accurate should the measurements be (i.e., find  $\sigma$ ) if we want the estimate to be within 10% of the true value with 99% probability?
4. Repeat (3) if  $\rho = 0$ .

**3-2 Sensitivity to incorrect prior variance.** The random variable  $x$  with prior mean  $\bar{x}$  and variance  $\sigma_0^2$  is measured via

$$z = x + w$$

where  $w$  is zero mean, with variance  $\sigma^2$ , and independent of  $x$ .

1. Write the *linear* MMSE estimator  $\hat{x}$  of  $x$  in terms of  $z$  and the MSE  $\sigma_1^2$  associated with this estimator.
2. Write the estimate  $x^*$  of  $x$  as above but under the *incorrect* assumption that the prior variance is  $\sigma_p^2$ .
3. Find the actual MSE,  $\sigma_a^2$ , associated with (2), and the MSE  $\sigma_c^2$  computed by the estimator in (2).
4. Let  $\sigma_p^2 = s\sigma_0^2$ . Assuming  $\sigma_0^2 = \sigma^2 = 2$ , evaluate the computed and the actual MSE for  $s = 0.5, 1, 2, 10, 20$  and compare with the best linear estimator.
5. Indicate the limits as  $s \rightarrow 0$  and  $s \rightarrow \infty$  and interpret the results.

**3-3 Fisher information matrix in the LS problem.** Derive the FIM for (3.4.1-10).

**3-4 Fisher information matrix in bearings-only target localization.** Prove (3.7.4-4).

**3-5 Passive localization with direction cosine measurements.** Rework the problem of Section 3.7 with measurements

$$h(k, x) = \frac{\xi - \xi_p}{(\xi - \xi_p)^2 + (\eta - \eta_p)^2}$$

**3-6 Covariance of the residual.** Prove (3.4.2-16).

**3-7 Noise correlated to prior error.** Consider the random variable  $x$  with

$$Ex = \bar{x} \quad \text{var}(x) = \sigma_0^2$$

The observation

$$z = x + w$$

is made, where  $w$  is a random variable with

$$Ew = 0 \quad \text{var}(w) = \sigma_w^2 \quad \text{cov}(x, w) = \rho\sigma_0\sigma_w$$

1. Find the LMMSE estimate of  $x$  given  $z$ .

2. Find the corresponding MMSE.
3. Find  $\rho$  that minimizes the MMSE.
4. For what value of  $\rho$  does this problem have no solution? Explain.

**3-8 A measurement that can be noise only.** Consider the random variable  $x \sim (\bar{x}, \sigma_x^2)$ . The observation

$$z = \alpha x + w$$

is made, where  $\alpha$  is a random variable with

$$P\{\alpha = 1\} = p_1 \quad P\{\alpha = 0\} = 1 - p_1$$

(i.e., the measurement can be only noise) and  $w \sim (0, \sigma_w^2)$ . The random variables  $x$ ,  $\alpha$  and  $w$  are mutually independent.

1. Find the LMMSE estimate of  $x$  in terms of  $z$  and the associated MSE. Evaluate for  $\bar{x} = 1$ ,  $\sigma_x^2 = 1$ ,  $p_1 = 1/2$ ,  $\sigma_w^2 = 1$ .
2. Find the (optimal) MMSE estimate of  $x$  in terms of  $z$  and the associated covariance. Assume that  $x$  and  $w$  are Gaussian with the same moments as before. Evaluate for the same numbers and compare.

**3-9 LMMSE estimation for vectors.** Let

$$y = Ax_1 + Bx_2$$

$$z = Cy + Dw$$

be vectors of suitable dimensions, with the matrices  $A$ ,  $B$ ,  $C$ ,  $D$  known,

$$x_i \sim (\bar{x}_i, P_{ii}), \quad i = 1, 2$$

(i.e., distributed with the indicated mean and covariance)

$$\text{cov}(x_1, x_2) = P_{12}$$

and

$$w \sim (\bar{w}, P_w)$$

orthogonal on  $x_i$ .

1. Find the LMMSE estimate of  $y$  in terms of  $z$
2. Find the associated matrix MSE.

**3-10 Velocity estimation.** Find an approximate lower bound on the estimation accuracy of the velocity of an object whose position is measured every 10 s over 70 s (8 observations) with a zero-mean noise with  $\sigma = 64$  ft. Express the results in ft/min (the calculation can be done by hand with an accuracy of a few percent).

- 3-11 A position estimation variance.** An object is moving in one dimension with unknown constant velocity. Its position is observed at equal sampling intervals with an additive zero-mean noise with variance  $\sigma^2$ .

1. Indicate the variance of its last position estimate after  $n$  measurements.

*Hint:* The result is (well) hidden somewhere in the text.

2. Now assume the velocity is known perfectly, but only the position is unknown and observed as before. What is the last position estimate variance after  $n$  measurements?
3. Show that the ratio of the position estimation variance with unknown constant velocity to the one with perfectly known velocity is  $4(n + 0.5)/(n - 1)$ .

- 3-12 Estimation from indirect observations.** Given the scalar random variables  $x_i$ ,  $i = 1, 2, 3$  jointly normal with means  $\mu_i$  and covariances  $\sigma_{ij}^2$ . Let

$$y_1 = x_2 + x_3$$

$$y_2 = x_2 - x_3$$

Find

1.  $p(x_1|y_1)$
2.  $p(x_1|y_2)$

*Hint:* There is no need to derive the densities. Use the fundamental equations of linear estimation.

- 3-13 Estimation with quadratic observations.** Consider the scalar random variable

$$x \sim \mathcal{N}(\bar{x}, P_{xx})$$

$$z = x + w$$

with

$$w \sim \mathcal{N}(0, P_{ww}) \quad E[xw] = 0$$

and

$$y = z^2$$

Find the LMMSE estimate of  $x$  in terms of  $y$  and the associated MSE.

- 3-14 Constant altitude trajectory estimation from range-only observations.** An object at unknown location  $x_0, y_0$  at  $t_0$  is moving in a plane with unknown constant velocity  $\dot{x}_0$ . It is known that  $\dot{y}(t) = 0 \quad \forall t$ .

Measurements  $r(t_k)$  of the range (distance to the object) are made at  $t_k$ ,  $k = 0, 1, \dots, N$ , by a sensor at the origin with additive Gaussian zero-mean white noises with s.d.  $\sigma_r$ .

- (i) Formulate the ML estimation problem for the parameter vector defining the motion.
- (ii) Provide all the expressions needed (by a programmer) to evaluate the CRLB for this problem.

- 3-15 Nonlinear L.S.** Consider the nonlinear LS problem in Subsection 3.4.4. Assume that now the sensor measures the range  $r(i)$  to the target at times  $i = 1, 2, 3$ . That is,

$$z = \begin{bmatrix} r(1) \\ r(2) \\ r(3) \end{bmatrix}$$

with

$$r(i) = \sqrt{(\eta - \eta_p(i))^2 + (\xi - \xi_p(i))^2} + v_r(i) \quad i = 1, 2, 3$$

where  $v_r \sim \mathcal{N}[0, (40\text{m})^2]$ .

1. Derive and implement the ILS estimator for the two scenarios shown in Figs. 3.4.4-2 and 3.4.4-3.
2. Derive and implement the ML estimator with Newton-Raphson search for the same scenarios.
3. Derive the CRLB and compare it with the performances of the ILS and ML estimators.
4. Evaluate the GDOP of the estimators for both scenarios.

Note that this problem represents a simplified version of the **Global Positioning System (GPS)**. See Chapter 12 for a detailed description of GPS.

- 3-16 LMMSE estimator.** Consider the random vectors  $x_i \sim (\bar{x}_i, P_{ii})$ ,  $\text{cov}[x_i, x_j] = P_{ij}$ ,  $i = 1, 2, 3$ . Let

$$z = x_1 + x_2 - x_3$$

Find the LMMSE estimator of each  $x_i$  in terms of  $z$  and the corresponding MSE errors.

- 3-17 Prediction with unequal sampling times and variances.** Let  $x = x_0 + v_0 t$ . Two measurements  $z_i = x(t_i) + w_i$ ,  $i = 1, 2$ , are made with zero-mean uncorrelated noises with (different) variances  $\sigma_i^2$ .

1. Find the estimates of  $x_0$ ,  $v_0$ .
2. Find the covariance matrix of their estimation errors.
3. Find the prediction of  $x(t_3)$ .
4. Find the variance of the error in 3.

*Note:* The sampling times and the noise variances are arbitrary.

### 3.8.3 PROJECT: An Interactive Program for Bearings-Only Target Localization

The following project is meant to develop a user-friendly interactive program for **bearings-only target localization**.

Set up an interactive program BEAR.EXE that accepts the following user-specified inputs for the problem of Section 3.7:

- A1. True value of the target parameter  $x$  corresponding to  $t = 0$ .
- A2. Sampling period  $T$ .
- A3. Number of samples  $n$ .
- A4. Measurement noise variance  $r$ .
- A5. Platform motion — initial position, initial velocity.
- A6. Platform maneuver — as velocities in each coordinate with starting and ending time ( $k_1$  and  $k_2$ , with the velocities being in effect from  $k_1T$  to  $k_2T$ ).
- A7. An initial estimate of the parameter to start the minimization. A set of default input values should be readable in from file BEAR.DEF.

The program should carry out the following:

- B1. Evaluation of the Fisher information matrix corresponding to the true target parameter vector. Based on this, indicate the observability by stating whether the FIM is invertible.
- B2. If the inverse can be obtained, it should be displayed.
- B3. The target position uncertainty for each sampling time should be computed and the corresponding covariance matrices displayed.
- B4. Generate noisy bearing measurements for the configuration defined.
- B5. The MLE of the target parameter should be obtained based on the noisy measurements using a minimization algorithm (Newton-Raphson or quasi-Newton recommended) starting from the initial estimate specified as part of the input and the result will be displayed.
- B6. Repeat B1–B3 for the estimated parameter.
- B7. The fitting error (sum of the squares of the residuals) should be evaluated and displayed together with its number of degrees of freedom.
- B8. The “normalized estimation error squared” (NEES) for the parameter vector of interest,  $\tilde{x}'J\tilde{x}$ , should be calculated and it should be indicated whether it is within its probability limit (e.g., 95%).

Another program, DBEAR.EXE, should display graphically the position uncertainty ellipses for the target, at user-specified times, centered at the estimate and their size according to the covariance matrices from B6. The ellipses will be of user-specified size (number of sigmas).

Document the algorithm that takes a  $2 \times 2$  symmetric positive definite matrix

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{12} & P_{22} \end{bmatrix}$$

and yields the “ $g$ -sigma” ellipse corresponding to it

$$x'P^{-1}x = g^2$$



# Chapter 4

## LINEAR DYNAMIC SYSTEMS WITH RANDOM INPUTS

### 4.1 INTRODUCTION

#### 4.1.1 Outline

This chapter deals with the modeling of linear dynamic systems excited by random inputs, called *noise*.<sup>1</sup> Continuous-time systems are discussed in Section 4.2. Section 4.3 deals with discrete-time systems.

The state-space models for continuous time and discrete time are presented, and it is shown how the latter can be derived from the former by discretization. The state-space model directly defined in discrete time is also discussed.

The Markov property of the state of a linear system driven by white noise is discussed and used to obtain the propagation equations for the mean and covariance of the state.

The power spectral density (the Fourier transform of the autocorrelation function) of the output of a linear system is related to the state space representation via the transfer function, and it is shown how its factorization makes it possible to prewhiten an autocorrelated random process or sequence.

#### 4.1.2 Linear Stochastic Systems — Summary of Objectives

Present the state-space models for

- continuous-time linear stochastic systems
- discrete-time linear stochastic systems

---

<sup>1</sup>According to a former program manager at a major Federal research agency, noise is beneficial — it lubricates the system. One can add to this that it also provides opportunities for research. More importantly, it was found via psychological experiments that people will lose their sanity without noise, which is particularly true for experts on stochastic systems (especially those dealing with estimation and tracking).

and the connection between them.

Discuss the implications of the Markov property.

Derive the propagation equations for the mean and covariance of the state of a linear system driven by white noise.

Frequency domain approach — connect the power spectral density with the state space representation.

Show how spectral factorization can be used to prewhiten an autocorrelated random process.

## 4.2 CONTINUOUS-TIME LINEAR STOCHASTIC DYNAMIC SYSTEMS

### 4.2.1 The Continuous-Time State-Space Model

The **state-space representation** of *continuous-time linear stochastic systems* can be written as

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) + D(t)\tilde{v}(t) \quad (4.2.1-1)$$

where

$x$  is the **state vector** of dimension  $n_x$ ,

$u$  is the **input vector** (control) of dimension  $n_u$ ,

$\tilde{v}$  is the (continuous-time) input disturbance or **process noise**, also called **plant noise**, a vector of dimension  $n_v$ ,

$A$ ,  $B$ , and  $D$  are known matrices of dimensions  $n_x \times n_x$ ,  $n_x \times n_u$ , and  $n_x \times n_v$ , respectively;

$A$  is called the **system matrix**,

$B$  is the (continuous-time) **input gain**,

$D$  is the (continuous-time) **noise gain**.

Equation (4.2.1-1) is known as the **dynamic equation** or the **plant equation**. The state  $x(t)$  is a (linear) **diffusion process** subject to a **drift** due to  $u(t)$ .

The output of the system is, in general, a vector of dimension  $n_z$

$$z(t) = C(t)x(t) + \tilde{w}(t) \quad (4.2.1-2)$$

where

$\tilde{w}$  is the (unknown) output disturbance or **measurement noise**, and

$C$  is a known  $n_z \times n_x$  matrix, called the **measurement matrix**.

Equation (4.2.1-2) is known as the **output equation** or the **measurement equation**.

In the absence of the disturbances  $\tilde{v}$  and  $\tilde{w}$ , that is, in the deterministic case, given the initial condition  $x(t_0)$  and the input **function in the interval**  $[t_0, t]$  denoted as

$$u_{[t_0, t]} \triangleq \{u(\tau), t_0 \leq \tau \leq t\} \quad (4.2.1-3)$$

one can compute the future output at any time  $t > t_0$

$$z(t) = z[x(t_0), u_{[t_0, t]}, t, t_0] \quad (4.2.1-4)$$

The **state** of a deterministic system is defined as the smallest vector that *summarizes the past of the system*.

Any linear differential equation that describes an input-output relationship can be put in the form of a first-order vector differential equation as in (4.2.1-1). For example, an  $n$ th-order scalar differential equation can be rewritten as a first-order differential equation for an  $n$ -vector — that is,  $n$  first-order equations — by a suitable definition of state variables.

The initial conditions of the  $n$ th-order differential equation can be taken as state variables or any invertible linear transformation of them.

In the stochastic case, as will be discussed in detail in Chapter 10, the pdf of the deterministic state vector of the system *summarizes the past in a probabilistic sense*. This requires that the process noise be *white*. Then the pdf of the state vector is called the **information state**.

In the stochastic case, the noises are usually assumed to be

1. zero-mean,
2. white, and
3. mutually independent

stochastic processes. If the noise is not zero mean, its mean (if known) can be taken as a known input.

## 4.2.2 Solution of the Continuous-Time State Equation

The state equation (4.2.1-1) has the following solution:

$$x(t) = F(t, t_0)x(t_0) + \int_{t_0}^t F(t, \tau)[B(\tau)u(\tau) + D(\tau)\tilde{v}(\tau)]d\tau \quad (4.2.2-1)$$

where  $x(t_0)$  is the initial state and  $F(t, t_0)$  is the **state transition matrix** from  $t_0$  to  $t$ .

The transition matrix has the following properties:

$$\frac{dF(t, t_0)}{dt} = A(t)F(t, t_0) \quad (4.2.2-2)$$

$$F(t_2, t_0) = F(t_2, t_1)F(t_1, t_0) \quad \forall t_1 \quad (4.2.2-3)$$

$$F(t, t) = I \quad (4.2.2-4)$$

The last two imply that

$$F(t, t_0) = F(t_0, t)^{-1} \quad (4.2.2-5)$$

The transition matrix has, in general, no explicit form, unless the following *commutativity property* is satisfied:

$$A(t) \int_{t_0}^t A(\tau) d\tau = \int_{t_0}^t A(\tau) d\tau A(t) \quad (4.2.2-6)$$

Then (and only then)

$$F(t, t_0) = e^{\int_{t_0}^t A(\tau) d\tau} \quad (4.2.2-7)$$

Condition (4.2.2-6) is satisfied for time-invariant systems or diagonal  $A(t)$ .

For a time-invariant system, assuming  $t_0 = 0$ , one has

$$F(t) \triangleq F(t, 0) = e^{At} \quad (4.2.2-8)$$

## Evaluation of the Transition Matrix

Some of the computational methods for the evaluation of the matrix  $e^{At}$  are briefly presented below.

1. Infinite series method:

$$e^{At} = \sum_{k=0}^{\infty} \frac{(At)^k}{k!} = I + At + \frac{A^2 t^2}{2} + \dots \quad (4.2.2-9)$$

where  $I$  is the identity matrix of the same dimension  $n \times n$  as  $A$ . This is a numerical method and it requires series truncation (unless a closed-form expression can be found for each term).

2. Laplace transform method:

$$e^{At} = \mathcal{L}^{-1}\{(sI - A)^{-1}\} \quad (4.2.2-10)$$

where  $\mathcal{L}^{-1}$  is the inverse Laplace transform. This is practical if one can find a closed-form expression of the required matrix inverse above.

3. *Interpolating polynomial* method. Compute the eigenvalues  $\lambda_i$  of  $A$ ,  $i = 1, \dots, n_e$ , where  $n_e$  is the number of *distinct eigenvalues*, with multiplicities  $m_i$ , and

$$\sum_{i=1}^{n_e} m_i = n \quad (4.2.2-11)$$

Then find a polynomial of degree  $n - 1$

$$g(\lambda) = \sum_{k=0}^{n-1} g_k \lambda^k \quad (4.2.2-12)$$

which is equal to  $e^{\lambda t}$  on the spectrum of  $A$ , that is,

$$\frac{d^j}{d\lambda^j} g(\lambda)|_{\lambda=\lambda_i} = \frac{d^j}{d\lambda^j} e^{\lambda t}|_{\lambda=\lambda_i} \quad i = 1, \dots, n_e, \quad j = 0, \dots, m_i - 1 \quad (4.2.2-13)$$

Then

$$e^{At} = g(A) \quad (4.2.2-14)$$

### Example — Coordinated Turn

Consider an object moving in a plane with **constant speed** (the *magnitude* of the velocity vector) and turning with a **constant angular rate** (i.e., executing a **coordinated turn** in aviation language).

The equations of motion in the plane  $(\xi, \eta)$  in this case are

$$\ddot{\xi} = -\Omega\dot{\eta} \quad \ddot{\eta} = \Omega\dot{\xi} \quad (4.2.2-15)$$

where  $\Omega$  is the constant angular rate ( $\Omega > 0$  implies a counterclockwise turn).<sup>2</sup>

The state space representation of the above with the state vector

$$x \triangleq [\xi \ \dot{\xi} \ \eta \ \dot{\eta}]' \quad (4.2.2-16)$$

is

$$\dot{x} = Ax \quad (4.2.2-17)$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\Omega \\ 0 & 0 & 0 & 1 \\ 0 & \Omega & 0 & 0 \end{bmatrix} \quad (4.2.2-18)$$

It can be easily shown that the eigenvalues of  $A$  are 0, 0, and  $\pm\Omega j$ .

It can be shown, using one of the techniques discussed earlier for evaluating the transition matrix, that for  $A$  given above one has

$$e^{At} = \begin{bmatrix} 1 & \frac{\sin \Omega t}{\Omega} & 0 & -\frac{1 - \cos \Omega t}{\Omega} \\ 0 & \cos \Omega t & 0 & -\sin \Omega t \\ 0 & \frac{1 - \cos \Omega t}{\Omega} & 1 & \frac{\sin \Omega t}{\Omega} \\ 0 & \sin \Omega t & 0 & \cos \Omega t \end{bmatrix} \quad (4.2.2-19)$$

This allows, among other things, easy generation of state trajectories for such turns (the position evolves along circular arcs). These turns are common for aircraft as well as other flying objects.

### 4.2.3 The State as a Markov Process

Assume the process noise entering the state equation (4.2.1-1) to be zero mean and *white*, that is, that  $\tilde{v}(t)$  is *independent* of  $\tilde{v}(\tau)$  for all  $t \neq \tau$ . In this case, the autocorrelation of  $\tilde{v}(t)$  is

$$E[\tilde{v}(t)\tilde{v}(\tau)'] = V(t)\delta(t - \tau) \quad (4.2.3-1)$$

---

<sup>2</sup>This is in accordance to the trigonometric convention. In the navigation convention, angles are measured clockwise (from North); thus a positive turn rate implies a clockwise turn.

The whiteness property of the process noise allows the preservation of the state's property of summarizing the past in the following sense: the pdf of the state at some time  $t$  conditioned on its values up to an earlier time  $t_1$  depends only on the last value  $x(t_1)$ :

$$p[x(t)|x_{[-\infty, t_1]}, u_{[t_1, t]}] = p[x(t)|x(t_1), u_{[t_1, t]}] \quad (4.2.3-2)$$

This follows from the complete unpredictability of the process noise due to its whiteness. Were the process noise autocorrelated ("colored"), (4.2.3-2) would not hold because states prior to  $t_1$  could be used to predict the process noise  $\tilde{v}_{[t_1, t]}$ , and thus  $x(t)$ , in some fashion.

This can be seen from the solution of the state equation

$$x(t) = F(t, t_1)x(t_1) + \int_{t_1}^t F(t, \tau)[B(\tau)u(\tau) + D(\tau)\tilde{v}(\tau)] d\tau \quad (4.2.3-3)$$

which indicates that  $x(t_1)$  summarizes the past, the input provides the known part of the state's evolution after  $t_1$ , and the last term above is the contribution of the process noise, which is *completely unpredictable*.

In other words, the state of a dynamic system driven by white noise is a *Markov process*.

#### 4.2.4 Propagation of the State's Mean and Covariance

Consider (4.2.1-1) with the known input  $u(t)$  and **nonstationary white process noise** with nonzero mean

$$E[\tilde{v}(t)] = \bar{v}(t) \quad (4.2.4-1)$$

and autocovariance function

$$E[(\tilde{v}(t) - \bar{v}(t))(\tilde{v}(\tau) - \bar{v}(\tau))'] = V(t)\delta(t - \tau) \quad (4.2.4-2)$$

The expected value of the state

$$\bar{x}(t) \triangleq E[x(t)] \quad (4.2.4-3)$$

evolves according to the (deterministic) differential equation

$$\dot{\bar{x}}(t) = A(t)\bar{x}(t) + B(t)u(t) + D(t)\bar{v}(t) \quad (4.2.4-4)$$

The above **propagation equation of the mean** follows from taking the expected value of (4.2.1-1) or differentiating the expected value of (4.2.2-1) with Leibniz' rule (see problem 4-1).

The covariance of the state

$$P_{xx}(t) \triangleq E[(x(t) - \bar{x}(t))(x(t) - \bar{x}(t))'] \quad (4.2.4-5)$$

has the expression

$$P_{xx}(t) = F(t, t_0)P_{xx}(t_0)F(t, t_0)' + \int_{t_0}^t F(t, \tau)D(\tau)V(\tau)D(\tau)'F(t, \tau)'d\tau \quad (4.2.4-6)$$

and evolves according to the differential equation, known as the **Lyapunov equation**,

$$\boxed{\dot{P}_{xx}(t) = A(t)P_{xx}(t) + P_{xx}(t)A(t)' + D(t)V(t)D(t)'} \quad (4.2.4-7)$$

This is the **propagation equation of the covariance**, and it can be proven by evaluating the covariance of (4.2.2-1) using the whiteness of  $\tilde{v}(t)$  and differentiating the result (see problem 4-1).

### Example

Consider the time-invariant scalar case with  $A(t) = -a < 0$  (a stable system),  $B(t) = 0$ ,  $D(t) = 1$ ,  $\bar{v}(t) = \bar{v}$ ,  $V(t) = V$ . The mean evolves according to

$$\dot{\bar{x}}(t) = -a\bar{x}(t) + \bar{v} \quad (4.2.4-8)$$

which, with initial condition  $\bar{x}(0)$ , yields

$$\bar{x}(t) = e^{-at}\bar{x}(0) + \frac{\bar{v}}{a}(1 - e^{-at}) \quad (4.2.4-9)$$

The variance evolution equation is

$$\dot{P}(t) = -2aP(t) + V \quad (4.2.4-10)$$

which, with initial condition  $P(0)$ , has the solution

$$P(t) = e^{-2at}P(0) + \frac{V}{2a}(1 - e^{-2at}) \quad (4.2.4-11)$$

### Note

In general, the Lyapunov equation has no explicit solution beyond (4.2.4-6), which has to be evaluated numerically. For stable systems the steady-state solution can be obtained with efficient numerical techniques (see, e.g., [Chen84]).

## 4.2.5 Frequency Domain Approach

Consider the *time-invariant* system driven by noise only

$$\dot{x}(t) = Ax(t) + D\tilde{v}(t) \quad (4.2.5-1)$$

where the noise is zero mean, *stationary*, and white, with autocorrelation function

$$R_{\tilde{v}\tilde{v}}(\tau) = E[\tilde{v}(t + \tau)\tilde{v}(t)'] = V\delta(\tau) \quad (4.2.5-2)$$

and with output

$$z(t) = Cx(t) \quad (4.2.5-3)$$

If the system is *stable* (i.e., all the eigenvalues of the system matrix  $A$  are in the left half-plane), then its output becomes a stationary process (when the transient period is over). The autocorrelation of the output is denoted as

$$R_{zz}(\tau) = E[z(t + \tau)z(t)'] \quad (4.2.5-4)$$

The power spectral density — **power spectrum** — of the process noise, which is the Fourier transform of its autocorrelation function, is

$$S_{\tilde{v}\tilde{v}}(\omega) = \int_{-\infty}^{\infty} R_{\tilde{v}\tilde{v}}(\tau) e^{-j\omega\tau} d\tau = V \quad (4.2.5-5)$$

It can be shown that the **power spectral density matrix of the output** — the Fourier transform of (4.2.5-4) — is

$$S_{zz}(\omega) = H(j\omega)S_{\tilde{v}\tilde{v}}(\omega)H(j\omega)^* = H(j\omega)VH(j\omega)^* \quad (4.2.5-6)$$

where the asterisks denote complex conjugate transpose and

$$H(j\omega) \triangleq C(j\omega I - A)^{-1}D \quad (4.2.5-7)$$

is the transfer function matrix of system (4.2.5-1) from the noise  $\tilde{v}$  to the output. Note that  $H(j\omega)$  is a **rational function** (ratio of polynomials).

## Spectral Factorization

Equation (4.2.5-6) leads to the following result: Given a **rational spectrum**, one can find a linear time-invariant system whose output, when driven by a stationary white noise, will have that spectrum.

The transfer function of such a system, called **prewhitening system** or **shaping filter**, is obtained by **factorization** of the desired spectrum into the product of a function with its complex conjugate. The first factor should correspond to a *causal and stable system*.

### Example — Exponentially Decaying Autocorrelation

Consider the scalar stochastic process with **exponentially decaying autocorrelation** or **Ornstein-Uhlenbeck process**

$$R_{zz}(\tau) = \sigma^2 e^{-\alpha|\tau|} \quad \alpha > 0 \quad (4.2.5-8)$$

The spectrum corresponding to the above is

$$S_{zz}(\omega) = \sigma^2 \frac{2\alpha}{\alpha^2 + \omega^2} \quad (4.2.5-9)$$

Factorization of this spectrum according to (4.2.5-6) yields

$$H(j\omega) = \frac{1}{\alpha + j\omega} \quad (4.2.5-10)$$

$$V = 2\alpha\sigma^2 \quad (4.2.5-11)$$

The state equation corresponding to (4.2.5-10) is

$$\dot{x}(t) = -\alpha x(t) + \tilde{v}(t) \quad (4.2.5-12)$$

with output

$$z(t) = x(t) \quad (4.2.5-13)$$

and process noise autocorrelation

$$E[\tilde{v}(t + \tau)\tilde{v}(t)] = 2\alpha\sigma^2\delta(\tau) \quad (4.2.5-14)$$

Note that the transfer function in (4.2.5-10) corresponds to a stable causal system, while its complex conjugate would represent an unstable noncausal system.

## 4.3 DISCRETE-TIME LINEAR STOCHASTIC DYNAMIC SYSTEMS

### 4.3.1 The Discrete-Time State-Space Model

In the **state space representation** of **discrete-time systems**, it is assumed that the input is piecewise constant, that is,

$$u(t) = u(t_k) \quad t_k \leq t < t_{k+1} \quad (4.3.1-1)$$

Then the state at sampling time  $t_{k+1}$  can be written, from (4.2.2-1), in terms of the state at  $t_k$  as

$$x(t_{k+1}) = F(t_{k+1}, t_k)x(t_k) + G(t_{k+1}, t_k)u(t_k) + v(t_k) \quad (4.3.1-2)$$

where  $F$  is the (**state**) **transition matrix** of the system,  $G$  is the **discrete-time gain** through which the input, assumed to be constant over a sampling period, enters the system and  $v(t_k)$  is the **discrete-time process noise**.

For a **time-invariant** continuous-time system sampled at arbitrary times, the transition matrix is

$$F(t_{k+1}, t_k) = F(t_{k+1} - t_k) = e^{(t_{k+1} - t_k)A} \triangleq F(k) \quad (4.3.1-3)$$

the input gain is

$$G(t_{k+1}, t_k) = \int_{t_k}^{t_{k+1}} e^{(t_{k+1}-\tau)A} B d\tau \triangleq G(k) \quad (4.3.1-4)$$

and the discrete-time process noise relates to the continuous-time noise as

$$v(t_k) = \int_{t_k}^{t_{k+1}} e^{(t_{k+1}-\tau)A} D \tilde{v}(\tau) d\tau \triangleq v(k) \quad (4.3.1-5)$$

Equations (4.3.1-3) to (4.3.1-5) introduce the simplified index-only notation for discrete-time systems, to be used (most of the time) in the sequel.

With the zero-mean and white assumption on  $\tilde{v}(t)$ , as in (4.2.3-1), it follows that

$$\boxed{E[v(k)] = 0} \quad (4.3.1-6)$$

$$\boxed{E[v(k)v(j)'] = Q(k)\delta_{kj}} \quad (4.3.1-7)$$

where  $\delta_{kj}$  is the Kronecker delta function. The covariance of the discrete-time process noise is given by

$$Q(k) = \int_{t_k}^{t_{k+1}} e^{(t_{k+1}-\tau)A} D V(\tau) D' e^{(t_{k+1}-\tau)A'} d\tau \quad (4.3.1-8)$$

The proof of (4.3.1-8) is given at the end of this subsection.

The (dynamic) model for *discrete-time linear stochastic systems* can be written with the simplified index-only time notation as

$$\boxed{x(k+1) = F(k)x(k) + G(k)u(k) + v(k)} \quad (4.3.1-9)$$

where the input is assumed known along with the matrices  $F(k)$  and  $G(k)$  and the process noise  $v(k)$  is a zero-mean, white random sequence with covariance matrix  $Q(k)$ . Any (known) nonzero mean of the process noise  $v$  can be incorporated into the input.

The *discrete-time measurement equation* is, with a similar notation,

$$\boxed{z(k) = H(k)x(k) + w(k)} \quad (4.3.1-10)$$

where  $H(k)$  is the **measurement matrix** and  $w(k)$  is the **measurement noise** — a random sequence with moments

$$\boxed{E[w(k)] = 0} \quad (4.3.1-11)$$

$$\boxed{E[w(k)w(j)'] = R(k)\delta_{kj}} \quad (4.3.1-12)$$

The measurement given by (4.3.1-10) represents a “short-term” integration, during which the state is assumed to be constant.

Note that (4.3.1-9) and (4.3.1-10) describe a *time-varying discrete-time system*.

The process and measurement noise sequences are (usually) assumed uncorrelated, that is,

$$E[v(k)w(j)'] = 0 \quad \forall k, j \quad (4.3.1-13)$$

In some cases it is convenient to define a **direct discrete-time model** rather than a discretized version of a continuous-time model. In such cases the process noise, also modeled as white, enters through a **noise gain**, denoted as  $\Gamma(k)$ . Then (4.3.1-9) is replaced by

$$\boxed{x(k+1) = F(k)x(k) + G(k)u(k) + \Gamma(k)v(k)} \quad (4.3.1-14)$$

In this case the process noise covariance  $Q(k)$  is defined directly. This will be discussed in more detail in Chapter 6.

### Derivation of the Covariance of the Discretized Process Noise

$$\begin{aligned} E[v(k)v(j)'] &= E \left\{ \int_{t_k}^{t_{k+1}} e^{(t_{k+1}-\tau_1)A} D\tilde{v}(\tau_1) d\tau_1 \left[ \int_{t_j}^{t_{j+1}} e^{(t_{j+1}-\tau_2)A} D\tilde{v}(\tau_2) d\tau_2 \right]' \right\} \\ &= E \left\{ \int_{t_k}^{t_{k+1}} \int_{t_j}^{t_{j+1}} e^{(t_{k+1}-\tau_1)A} D\tilde{v}(\tau_1) \tilde{v}(\tau_2)' D'e^{(t_{j+1}-\tau_2)A'} d\tau_1 d\tau_2 \right\} \\ &= \int_{t_k}^{t_{k+1}} \int_{t_j}^{t_{j+1}} e^{(t_{k+1}-\tau_1)A} DE[\tilde{v}(\tau_1)\tilde{v}(\tau_2)'] D'e^{(t_{j+1}-\tau_2)A'} d\tau_1 d\tau_2 \\ &= \int_{t_k}^{t_{k+1}} \int_{t_j}^{t_{j+1}} e^{(t_{k+1}-\tau_1)A} DV(\tau_1) \delta(\tau_1 - \tau_2) D'e^{(t_{j+1}-\tau_2)A'} d\tau_1 d\tau_2 \\ &= \int_{t_k}^{t_{k+1}} e^{(t_{k+1}-\tau_1)A} DV(\tau_1) D'e^{(t_{k+1}-\tau_1)A'} d\tau_1 \delta_{kj} \end{aligned} \quad (4.3.1-15)$$

### 4.3.2 Solution of the Discrete-Time State Equation

Using (4.3.1-9) for time  $k$  and substituting  $x(k-1)$  yields

$$\begin{aligned} x(k) &= F(k-1)x(k-1) + G(k-1)u(k-1) + v(k-1) \\ &= F(k-1)[F(k-2)x(k-2) + G(k-2)u(k-2) + v(k-2)] \\ &\quad + G(k-1)u(k-1) + v(k-1) \\ &= F(k-1)F(k-2)x(k-2) + F(k-1)[G(k-2)u(k-2) \\ &\quad \cdot + v(k-2)] + G(k-1)u(k-1) + v(k-1) \end{aligned} \quad (4.3.2-1)$$

Repeating the above leads to

$$x(k) = \left[ \prod_{j=0}^{k-1} F(k-1-j) \right] x(0) + \sum_{i=0}^{k-1} \left[ \prod_{j=0}^{k-i-2} F(k-1-j) \right] [G(i)u(i) + v(i)] \quad (4.3.2-2)$$

The notation for the product of matrices used above is

$$\prod_{j=j_1}^{j_2} F(j) \triangleq F(j_1)F(j_1+1) \dots F(j_2) \quad (4.3.2-3)$$

If the upper index in (4.3.2-3) is smaller than the lower index, the result is taken as the identity matrix.

Note that

$$\prod_{j=0}^{k-i-1} F(k-1-j) = F(k-1)F(k-2) \dots F(i) = F(t_k, t_i) \quad (4.3.2-4)$$

is the transition matrix from sampling time  $i$  to sampling time  $k$ .

If the discrete-time system is *time-invariant*, that is,

$$F(k) = F, \quad G(k) = G \quad \forall k \quad (4.3.2-5)$$

then (4.3.2-2) becomes

$$x(k) = F^k x(0) + \sum_{i=0}^{k-1} F^{k-i-1} [Gu(i) + v(i)] \quad (4.3.2-6)$$

### 4.3.3 The State as a Markov Process

If the discrete-time representation is obtained by discretizing a continuous-time system with white process noise, the resulting discrete-time process noise is a white sequence.

Similarly to the continuous-time case, one has, following (4.3.2-1),

$$x(k) = [\prod_{j=0}^{k-l-1} F(k-1-j)]x(l) + \sum_{i=l}^{k-1} [\prod_{j=0}^{k-i-2} F(k-1-j)][G(i)u(i) + v(i)] \quad (4.3.3-1)$$

Thus, since  $v(i)$ ,  $i = l, \dots, k-1$ , are independent of

$$X^l \triangleq \{x(j)\}_{j=0}^l \quad (4.3.3-2)$$

which depend only on  $v(i)$ ,  $i = 0, \dots, l-1$ , one has

$$p[x(k)|X^l, U^{k-1}] = p[x(k)|x(l), U_l^{k-1}] \quad \forall k > l \quad (4.3.3-3)$$

where

$$U_l^{k-1} \triangleq \{u(j)\}_{j=l}^{k-1} \quad (4.3.3-4)$$

Thus, the state vector is a *Markov process*, or, more correctly, a *Markov sequence*.

As in the continuous-time case, this follows from the complete unpredictability of the process noise due to its whiteness. Were the process noise autocorrelated ("colored"), (4.3.3-3) would not hold because states prior to time  $l$  could be used to predict the process noises  $v(i)$ ,  $i = l, \dots, k-1$ , and thus  $x(k)$ , in some fashion.

### 4.3.4 Propagation of the State's Mean and Covariance

Consider (4.3.1-14), repeated below for convenience

$$x(k+1) = F(k)x(k) + G(k)u(k) + \Gamma(k)v(k) \quad (4.3.4-1)$$

with the known input  $u(k)$  and the process noise  $v(k)$  white, but for the sake of generality, *nonstationary* with nonzero mean:

$$E[v(k)] = \bar{v}(k) \quad (4.3.4-2)$$

$$\text{cov}[v(k), v(j)] = E[[v(k) - \bar{v}(k)][v(j) - \bar{v}(j)']] = Q(k)\delta_{kj} \quad (4.3.4-3)$$

Then the expected value of the state

$$\bar{x}(k) \triangleq E[x(k)] \quad (4.3.4-4)$$

evolves according to the difference equation

$$\boxed{\bar{x}(k+1) = F(k)\bar{x}(k) + G(k)u(k) + \Gamma(k)\bar{v}(k)} \quad (4.3.4-5)$$

The above, which is the *propagation equation of the mean*, follows immediately by applying the expectation operator to (4.3.4-1).

The covariance of the state

$$P_{xx}(k) \triangleq E[[x(k) - \bar{x}(k)][x(k) - \bar{x}(k)']] \quad (4.3.4-6)$$

evolves according to the difference equation — the *covariance propagation equation*

$$\boxed{P_{xx}(k+1) = F(k)P_{xx}(k)F(k)' + \Gamma(k)Q(k)\Gamma(k)'} \quad (4.3.4-7)$$

This follows by subtracting (4.3.4-5) from (4.3.4-1), which yields

$$x(k+1) - \bar{x}(k+1) = F(k)[x(k) - \bar{x}(k)] + \Gamma(k)[v(k) - \bar{v}(k)] \quad (4.3.4-8)$$

It can be easily shown that multiplying (4.3.4-8) with its transpose and taking the expectation yields (4.3.4-7). The resulting cross-terms on the right-hand side vanish when the expectation is taken. This is due to the whiteness of the process noise, which causes  $x(k)$ , being a linear combination of the noises prior to  $k$ , to be *independent* of  $v(k)$ .

#### Example — Fading Memory Average

Consider the scalar system (also called “first-order Markov”)

$$x(k) = \alpha x(k-1) + v(k) \quad k = 1, \dots \quad (4.3.4-9)$$

with  $x(0) = 0$  and  $0 < \alpha < 1$ . Note the slight change in the time argument in comparison to (4.3.4-1).

For this system it can be easily shown directly, or, using (4.3.2-2), that its solution is

$$x(k) = \sum_{i=1}^k \alpha^{k-i} v(i) \quad k = 1, \dots \quad (4.3.4-10)$$

Normalizing the above by the sum of the coefficients yields

$$z(k) \triangleq \frac{x(k)}{\sum_{i=1}^k \alpha^{k-i}} \quad (4.3.4-11)$$

which is the **fading memory average**, or, **exponentially discounted average** of the sequence  $v(k)$ . The term fading memory average is sometimes used for (4.3.4-10), which is really a **fading memory sum** — without the normalization. Note that for  $\alpha = 1$ , (4.3.4-11) becomes the sample average.

If the input has constant mean

$$E[v(k)] = \bar{v} \quad (4.3.4-12)$$

then

$$x(k) \triangleq E[x(k)] = \bar{v} \sum_{i=1}^k \alpha^{k-i} = \bar{v} \frac{1 - \alpha^k}{1 - \alpha} \quad (4.3.4-13)$$

and

$$\lim_{k \rightarrow \infty} \bar{x}(k) = \frac{\bar{v}}{1 - \alpha} \quad (4.3.4-14)$$

It can be easily shown that

$$\bar{z}(k) \triangleq E[z(k)] = \bar{v} \quad \forall k \quad (4.3.4-15)$$

The use of the fading memory average is for the case of a “slowly” varying mean  $\bar{v}(k)$ . The larger weightings on the more recent values of  $v(k)$  allow “tracking” of its mean at the expense of the accuracy (larger variance) — see also problem 4-4.

### 4.3.5 Frequency Domain Approach

Consider the time-invariant system driven by noise only,

$$x(k+1) = Fx(k) + \Gamma v(k) \quad (4.3.5-1)$$

where the noise is assumed zero mean, stationary, and white, with autocorrelation

$$R_{vv}(k-l) = E[v(k)v(l)'] = Q\delta_{kl} \quad (4.3.5-2)$$

The output of the system is

$$z(k) = Hx(k) \quad (4.3.5-3)$$

The **power spectral density (spectrum)** of the discrete-time process noise is the **discrete-time Fourier transform (DTFT)**<sup>3</sup>

$$S_{vv}(e^{j\omega T}) = \sum_{n=-\infty}^{\infty} e^{-j\omega T n} R_{vv}(n) = Q \quad (4.3.5-4)$$

where  $\omega$  is the angular frequency and  $T$  the sampling period. Note the constant (flat) spectrum that characterizes a white noise.

If system (4.3.5-1) is **stable** (i.e., all the eigenvalues of  $F$  are *inside the unit circle*), then the state  $x(k)$  will also become a stationary random sequence. The spectrum of the output will be

$$S_{zz}(e^{j\omega T}) = \mathcal{H}(\zeta) S_{vv}(\zeta) \mathcal{H}(\zeta)^*|_{\zeta=e^{j\omega T}} \quad (4.3.5-5)$$

where the asterisk denotes complex conjugate transpose,  $\zeta$  is the (two-sided)  $z$ -transform variable, and

$$\mathcal{H}(\zeta) = H(\zeta I - F)^{-1} \Gamma \quad (4.3.5-6)$$

is the **discrete-time transfer function** of the system given by (4.3.5-1) and (4.3.5-3) from the process noise to the output. Note that the transfer function (4.3.5-6) is a **rational function**.

Using (4.3.5-4) in (4.3.5-5) yields the spectrum of the output as

$$S_{zz}(e^{j\omega T}) = \mathcal{H}(e^{j\omega T}) Q \mathcal{H}(e^{j\omega T})^* = \mathcal{H}(e^{j\omega T}) Q \mathcal{H}(e^{-j\omega T})' \quad (4.3.5-7)$$

which, being the product of two rational functions and a constant matrix, is also a rational function.

## Spectral Factorization

In view of (4.3.5-7), given a *rational spectrum*, one can find the linear time-invariant system which, driven by white noise, will have the output with the desired spectrum. Based on (4.3.5-7), the transfer function of such a system is obtained by **factorization** of the spectrum as follows:

$$S_{zz}(\zeta) = \mathcal{H}(\zeta) Q \mathcal{H}(\zeta^{-1})' \quad (4.3.5-8)$$

since, for

$$\zeta = e^{j\omega T} \quad (4.3.5-9)$$

one has

$$\zeta^* = \zeta^{-1} \quad (4.3.5-10)$$

The resulting transfer function  $\mathcal{H}(\zeta)$  specifies the **prewhitening system** or **shaping filter** for the sequence  $z(k)$ .

---

<sup>3</sup>The *DFT* is the transform based on a *finite number* of points and evaluated at the *same number* of sampled frequencies; it is usually obtained via the FFT.

**Remark**

The factor in (4.3.5-8) that corresponds to a *causal* and *stable* system is the one that is to be chosen for the transfer function of the prewhitening system.

**Example**

Consider the scalar sequence  $z(k)$  with mean zero and autocorrelation

$$R_{zz}(n) = \rho\delta_{n,-1} + \delta_{n,0} + \rho\delta_{n,1} \quad (4.3.5-11)$$

Its spectrum — the DTFT written as the two-sided  $z$ -transform — is

$$S_{zz}(\zeta) = 1 + \rho\zeta^{-1} + \rho\zeta \quad (4.3.5-12)$$

This can be factorized as

$$S_{zz}(\zeta) = \mathcal{H}(\zeta)\mathcal{H}(\zeta^{-1}) = (\beta_0 + \beta_1\zeta^{-1})(\beta_0 + \beta_1\zeta) \quad (4.3.5-13)$$

where the first factor is the transfer function of a causal system.

The equations for the coefficients  $\beta_0$  and  $\beta_1$  are

$$\beta_0^2 + \beta_1^2 = 1 \quad (4.3.5-14)$$

$$\beta_0\beta_1 = \rho \quad (4.3.5-15)$$

with solution

$$\beta_0 = \frac{1}{2}(\sqrt{1+2\rho} + \sqrt{1-2\rho}) \quad (4.3.5-16)$$

$$\beta_1 = \frac{1}{2}(\sqrt{1+2\rho} - \sqrt{1-2\rho}) \quad (4.3.5-17)$$

Thus the transfer function of the system is

$$\mathcal{H}(\zeta) = \beta_0 + \beta_1\zeta^{-1} \quad (4.3.5-18)$$

which corresponds to the following ***moving average (MA)*** or ***finite impulse response (FIR)*** system driven by unity-variance white noise

$$x(k) = \beta_0 v(k) + \beta_1 v(k-1) \quad (4.3.5-19)$$

$$z(k) = x(k) \quad (4.3.5-20)$$

Equations (4.3.5-19) and (4.3.5-20) specify the prewhitening system corresponding to (4.3.5-11).

## 4.4 SUMMARY

### 4.4.1 Summary of State Space Representation

*State* of a deterministic system — the smallest vector that *summarizes in full* its past.

*State equation* — a first-order differential or difference equation that describes the evolution in time (the dynamics) of the state vector.

*Markov process* — a stochastic process whose current state contains all the information about the probabilistic description of its future evolution.

A *state equation* driven by *white noise* yields a (vector) *Markov process*.

State of a stochastic system described by a Markov process — *summarizes probabilistically* its past.

All the above statements hold for linear as well as nonlinear systems.

A linear stochastic system's *continuous-time* representation as a differential equation driven by white noise can be written in *discrete-time* as a difference equation driven by a sequence of independent random variables — discrete-time white noise, which can be related to the continuous-time noise.

Alternatively, one can define directly a discrete-time state equation.

*Continuous-time white noise*: a random process with autocorrelation function a Dirac (impulse) delta function. If it is stationary, it has a spectrum (Fourier transform of the autocorrelation) that is constant (“flat”).

*Discrete-time white noise*: a random sequence with autocorrelation function being a Kronecker delta function. If it is stationary, it has a spectrum (DTFT of the autocorrelation) that is constant (“flat”).

The *unconditional mean and covariance* of the state of a linear stochastic system driven by white noise have been shown to evolve (“open loop”) according to linear differential or difference equations.

### 4.4.2 Summary of Prewhitening

For a linear time-invariant system, the frequency domain approach relates the power spectral density of the output to the one of the input via the transfer function.

The spectral density of the output of a stable linear time-invariant system is given by:

The spectrum of the input premultiplied by the system's transfer function (a rational function) and postmultiplied by its complex conjugate and transpose.

Given a rational spectrum, one can find a linear time-invariant system which, if driven by a stationary white noise, its output will have that spectrum:

The transfer function of such a *prewhitening system* is obtained by factoring the desired spectrum into the product of a function with its complex conjugate — this is called *spectral factorization*.

Due to the facts that

- Estimation results for dynamic systems, to be discussed in the sequel, are for Markov systems; and
- (Most) Markov processes (of interest) can be represented by linear time-invariant systems driven by white noise,

it is very important to find the “prewhitening system” for a given random process or sequence. This can be accomplished via spectral factorization.

## 4.5 NOTES AND PROBLEMS

### 4.5.1 Bibliographical Notes

The material on state space representation of stochastic systems is also discussed in, for example, [Sage71]. More on the continuous-time and discrete-time representations can be found, for example, in [Fortmann77] and [Chen84].

The relationship between the spectra of the output and input of a linear time-invariant system is proven in standard probability texts — for example, in [Papoulis84]. Spectral factorization is treated in more detail in [Anderson79].

### 4.5.2 Problems

#### 4-1 Mean and covariance of the state of a linear system.

1. Prove (4.2.4-4) by differentiating the expected value of (4.2.2-1) with Leibniz' rule

$$\frac{d}{dt} \int_{a(t)}^{b(t)} f(t, \tau) d\tau = \int_{a(t)}^{b(t)} \frac{\partial f(t, \tau)}{\partial t} d\tau + \frac{db(t)}{dt} f[t, b(t)] - \frac{da(t)}{dt} f[t, a(t)]$$

2. Prove (4.2.4-6).
3. Prove (4.2.4-7).

#### 4-2 Moments of the white noise response of a scalar linear system.

Consider the system

$$\dot{x}(t) = \alpha x(t) + \tilde{v}(t)$$

with  $\tilde{v}(t)$  white, with mean  $\bar{v}$  and variance  $q$ .

1. Find  $\alpha$  such that the mean of the resulting stationary process is  $\bar{x} = c$ . (*Hint:* This is the steady state of the differential equation of the mean.)
2. Using a time domain approach, find the autocorrelation of the resulting stationary process.

**4-3 Autocovariance of the state of a discrete-time system.**

- Find the autocovariance

$$V_{xx}(k, j) \triangleq E[(x(k) - \bar{x}(k))(x(j) - \bar{x}(j))']$$

in terms of  $P_{xx}(j)$  for system (4.3.1-14). Assume  $k > j$ .

- Indicate what happens for a stable linear time-invariant system for  $k \gg j$ . Justify.

**4-4 Fading memory average and effective window length.** Consider

$$y(k) = \alpha y(k-1) + (1-\alpha)v(k)$$

with  $y(0) = 0$  and  $0 < \alpha < 1$ .

- Write the solution for  $y(k)$ .
- Find the mean and variance of  $y(k)$  if  $v(k)$  is white with mean  $\bar{v}$  and variance  $\sigma^2$ .
- How does  $y(k)$  differ from  $z(k)$  in (4.3.4-11)?
- What is the “effective memory” (window length)  $N_e$  as a function of  $\alpha$ ? Determine  $\alpha$  for  $N_e = 10$ .

**4-5 Spectral factorization for prewhitening (shaping).** Given the scalar zero-mean random sequence  $x(k)$  with autocorrelation

$$R_{xx}(n) = E[x(k)x(k+n)] = \sigma^2 a^{|n|} \quad 0 < a < 1$$

- Find its spectrum.
- Factorize it to find the causal and stable linear system driven by white noise (prewhitening system or shaping filter) whose output has this autocorrelation.

**4-6 Autocovariance of the state of a system driven by nonstationary white noise.** Consider

$$\dot{x}(t) = Ax(t) + D\hat{v}(t)$$

$$E[x(t_0)] = \bar{x}(t_0)$$

$$E[(x(t_0) - \bar{x}(t_0))[x(t_0) - \bar{x}(t_0)]'] = P_{xx}(t_0)$$

$$E[\hat{v}(t)] = \bar{v}(t)$$

$$E[(\hat{v}(t) - \bar{v}(t))(\hat{v}(\tau) - \bar{v}(\tau))'] = Q(t)\delta(t - \tau)$$

Let

$$E[x(t)] \triangleq \bar{x}(t)$$

Find

$$V_{xx}(t, \tau) \triangleq E[(x(t) - \bar{x}(t))(x(\tau) - \bar{x}(\tau))']$$

**4-7 State prediction in a linear time-invariant discrete time system.**

1. Simplify (4.3.3-1) for a time-invariant system, that is,  $F(i) = F$ ,  $G(i) = G$ .
2. Find a closed-form solution, similar to (4.3.3-1), for the covariance (4.3.4-7) assuming  $F(k) = F$ ,  $\Gamma(k) = \Gamma$ ,  $Q(k) = Q$ .

**4-8 Coordinated turn transition matrix.**

1. Derive (4.2.2-15).
2. Derive (4.2.2-19).

**4-9 State equation for fading memory average.**

1. Find the time-invariant linear state equation that yields, for  $n \gg 1$ , the fading memory average

$$x(n) = \frac{\sum_{i=0}^n \alpha^{n-i} u(i)}{\sum_{i=0}^n \alpha^{n-i}} \quad 0 < \alpha < 1$$

2. With  $x(0) = 0$ ,  $E[u(i)] = \bar{u}$ ,  $\text{cov}[u(i), u(j)] = \sigma^2 \delta_{ij}$ , find the mean and variance of  $x(n)$
3. Verify that for  $n \gg 1$ , one has  $E[x(n)] = \bar{u}$
4. Find the asymptotic value of  $\text{var}[x(n)]$ . Is  $x(n)$  a consistent estimator of the parameter  $\bar{u}$ ?
5. Determine  $\alpha$  such that the standard deviation of the estimate of the parameter  $\bar{u}$  is  $0.1\sigma$ .

**4-10 Independence of measurements.**

1. Are the measurements in a standard noisy linear dynamic system independent?
2. Is there a way in which they are independent?

**4-11 Spectral factorization for a discrete-time system.** Consider the scalar zero-mean random sequence  $x(k)$  with autocorrelation

$$R_{xx}(n) = E[x(k)x(k+n)] = \sigma^2 a^{|n|} \quad 0 < a < 1 \quad n = \dots, -1, 0, 1, \dots$$

1. Find its spectrum.
2. Factorize it to find the causal and stable linear system driven by white noise (prewhitening system or shaping filter) whose output has this autocorrelation.
3. Write the state space equation of this system.

# Chapter 5

## STATE ESTIMATION IN DISCRETE-TIME LINEAR DYNAMIC SYSTEMS

### 5.1 INTRODUCTION

#### 5.1.1 Outline

This chapter extends the estimation concepts presented previously to the case of dynamic (time-varying) quantities. The estimation of the *state vector* of a *stochastic linear dynamic system* is considered.

The state estimator for discrete-time linear dynamic systems driven by white noise — the (discrete-time) **Kalman filter** — is introduced in Section 5.2 and its properties are discussed. The continuous-time case is considered later, in Chapter 9. An example that illustrates the discrete time Kalman filter is given in Section 5.3.

The issue of consistency<sup>1</sup> of a dynamic estimator, which is crucial for *evaluation of estimator optimality* in every implementation, is discussed in Section 5.4.

The initialization of estimators and practical ways to make it consistent are presented in Section 5.5.

#### 5.1.2 Discrete-Time Linear Estimation — Summary of Objectives

Extend the static estimation concepts to dynamic systems.

Derive the state estimator for discrete-time linear dynamic systems driven by white noise — the (discrete-time) Kalman filter.

Present the properties of the Kalman filter:

---

<sup>1</sup>Or, “how to keep the filter honest”; the origin of the preoccupation with this issue is summarized in problem 5-6.

- Optimality in the linear Gaussian case and best linear state estimator in the (linear) non-Gaussian case
- Whiteness of the innovations
- The role of the Riccati equation for the state covariance and its relationship with the CRLB
- Stability and steady-state
- Connection with the observability and controllability of the system

Introduce the *likelihood function of a filter*, to be used later in adaptive filtering.

Discuss the issues of

- Consistency of a dynamic estimator and *estimator evaluation*
- Initialization of estimators

and practical ways to make it consistent.

## 5.2 LINEAR ESTIMATION IN DYNAMIC SYSTEMS — THE KALMAN FILTER

### 5.2.1 The Dynamic Estimation Problem

Consider a discrete-time linear dynamic system described by a vector difference equation with additive white Gaussian noise that models “unpredictable disturbances.” The dynamic (plant) equation is

$$x(k+1) = F(k)x(k) + G(k)u(k) + v(k) \quad k = 0, 1, \dots \quad (5.2.1-1)$$

where  $x(k)$  is the  $n_x$ -dimensional state vector,  $u(k)$  is an  $n_u$ -dimensional **known input** vector (e.g., control or sensor platform motion), and  $v(k)$ ,  $k = 0, 1, \dots$ , is the sequence of zero-mean white Gaussian **process noise** (also  $n_x$ -vectors) with covariance

$$E[v(k)v(k)'] = Q(k) \quad (5.2.1-2)$$

The measurement equation is

$$z(k) = H(k)x(k) + w(k) \quad k = 1, \dots \quad (5.2.1-3)$$

with  $w(k)$  the sequence of zero-mean white Gaussian **measurement noise** with covariance

$$E[w(k)w(k)'] = R(k) \quad (5.2.1-4)$$

The matrices  $F$ ,  $G$ ,  $H$ ,  $Q$ , and  $R$  are assumed **known** and possibly time-varying. In other words, the system can be *time-varying* and the noises *non-stationary*. The initial state  $x(0)$ , in general unknown, is modeled as a *random variable*, Gaussian distributed with known mean and covariance. The two noise sequences and the initial state are assumed *mutually independent*.

The above constitutes the *linear Gaussian (LG) assumption*.

In the dynamic equation (5.2.1-1), the process noise term  $v(k)$  is sometimes taken as  $\Gamma(k)v(k)$  with  $v(k)$  an  $n_v$ -vector and  $\Gamma(k)$  a known  $n_x \times n_v$  matrix. Then the covariance matrix of the disturbance in the state equation, which is  $Q(k)$  if  $v(k)$  enters directly, is to be replaced by

$$E[[\Gamma(k)v(k)][\Gamma(k)v(k)]'] = \Gamma(k)Q(k)\Gamma(k)' \quad (5.2.1-5)$$

The linearity of (5.2.1-1) and (5.2.1-3) leads to the preservation of the Gaussian property of the state and measurements — this is a *Gauss-Markov process*.

The following notation will be used: The conditional mean

$$\hat{x}(j|k) \triangleq E[x(j)|Z^k] \quad (5.2.1-6)$$

where

$$Z^k \triangleq \{z(i), i \leq k\} \quad (5.2.1-7)$$

denotes the sequence of observations available at time  $k$ , is the

- **Estimate of the state** if  $j = k$  (also called filtered value)
- **Smoothed value of the state** if  $j < k$
- **Predicted value of the state** if  $j > k$

The *estimation error* is defined as

$$\tilde{x}(j|k) \triangleq x(j) - \hat{x}(j|k) \quad (5.2.1-8)$$

The *conditional covariance matrix* of  $x(j)$  given the data  $Z^k$  or the *covariance associated with the estimate* (5.2.1-6) is

$$P(j|k) \triangleq E[[x(j) - \hat{x}(j|k)][x(j) - \hat{x}(j|k)]'|Z^k] = E[\tilde{x}(j|k)\tilde{x}(j|k)'|Z^k] \quad (5.2.1-9)$$

## Remarks

The smoothed state has also been called recently the *retrodiction state* [Drummond93]. The term smoothing is, however, commonly used, even though retrodiction is the correct antonym of prediction. Sometimes the estimated state is called (incorrectly) the smoothed state.

Note that the *covariance of the state* is the same as the *covariance of the estimation error* — this is a consequence of the fact that the estimate is the conditional mean (5.2.1-6). (See also problem 5-3.)

It was shown earlier that the *MMSE criterion* for estimation leads to the *conditional mean* as the *optimal estimate*.

As discussed in Section 3.2, if two vectors are jointly Gaussian, then the probability density of one conditioned on the other is also Gaussian. Thus the conditional mean (5.2.1-6) will be evaluated using this previous result.

## The Estimation Algorithm

The estimation algorithm starts with the **initial estimate**  $\hat{x}(0|0)$  of  $x(0)$  and the associated **initial covariance**  $P(0|0)$ , assumed to be available. The second (conditioning) argument 0 stands for  $Z^0$ , the **initial information**. Practical procedures to obtain the initial estimate and initial covariance will be discussed later.

One cycle of the dynamic estimation algorithm — the **Kalman filter (KF)** — will thus consist of mapping the estimate

$$\hat{x}(k|k) \triangleq E[x(k)|Z^k] \quad (5.2.1-10)$$

which is the conditional mean of the state at time  $k$  (the “current stage”) given the observations up to and including time  $k$ , and the associated covariance matrix

$$P(k|k) = E[(x(k) - \hat{x}(k|k))(x(k) - \hat{x}(k|k))'|Z^k] \quad (5.2.1-11)$$

into the corresponding variables at the next stage, namely,  $\hat{x}(k+1|k+1)$  and  $P(k+1|k+1)$ .

This follows from the fact that a Gaussian random variable is *fully characterized* by its first two moments.

The values of past known inputs are subsumed in the conditioning, but (most of the time) will not be shown explicitly.

## 5.2.2 Dynamic Estimation as a Recursive Static Estimation

The recursion that yields the state estimate at  $k+1$  and its covariance can be obtained from the static estimation equations (3.2.1-7) and (3.2.1-8)

$$\hat{x} \triangleq E[x|z] = \bar{x} + P_{xz}P_{zz}^{-1}(z - \bar{z}) \quad (5.2.2-1)$$

$$P_{xx|z} \triangleq E[(x - \hat{x})(x - \hat{x})'|z] = P_{xx} - P_{xz}P_{zz}^{-1}P_{zx} \quad (5.2.2-2)$$

by the following substitutions, indicated below by “ $\rightarrow$ ”.

The *prior* (unconditional) expectations from the static case become *prior to the availability of the measurement at time  $k+1$*  in the dynamic case, that is, *given the data up to and including  $k$* .

The *posterior* (conditional) expectations become *posterior to obtaining the measurement at time  $k+1$* , that is, *given the data up to and including  $k+1$* .

The variable to be estimated is the state at  $k+1$

$$x \rightarrow x(k+1) \quad (5.2.2-3)$$

Its mean prior to  $k+1$  — the **(one-step) predicted state** — is

$$\bar{x} \rightarrow \bar{x}(k+1) \triangleq \hat{x}(k+1|k) \triangleq E[x(k+1)|Z^k] \quad (5.2.2-4)$$

Based on the observation (measurement)

$$z \rightarrow z(k+1) \quad (5.2.2-5)$$

with prior mean — the ***predicted measurement***

$$\bar{z} \rightarrow \bar{z}(k+1) \triangleq \hat{z}(k+1|k) \triangleq E[z(k+1)|Z^k] \quad (5.2.2-6)$$

one computes the estimate posterior to  $k+1$  — the ***updated state estimate*** (or, just the ***updated state***)

$$\hat{x} \rightarrow \hat{x}(k+1) \triangleq \hat{x}(k+1|k+1) \triangleq E[x(k+1)|Z^{k+1}] \quad (5.2.2-7)$$

### Note

Two notations have been used in (5.2.2-4) and (5.2.2-6): with single argument and with two arguments. The single-argument notation — with “overbar” for one-step predicted values and “hat” for updated value — is patterned after the static case. The two-argument notation, which clearly indicates the time index of the conditioning data, is more general and is the preferred one because it applies to smoothing and general prediction as well. Nevertheless, one can find both in the literature.

### The Covariances

The prior covariance matrix of the state variable  $x(k+1)$  to be estimated — the ***state prediction covariance*** or ***predicted state covariance*** — is

$$P_{xx} \rightarrow \bar{P}(k+1) \triangleq P(k+1|k) \triangleq \text{cov}[x(k+1)|Z^k] = \text{cov}[\tilde{x}(k+1|k)|Z^k] \quad (5.2.2-8)$$

with the last equality following from (5.2.1-9).

The (prior) covariance of the observation  $z(k+1)$  — the ***measurement prediction covariance*** — is

$$P_{zz} \rightarrow S(k+1) \triangleq \text{cov}[z(k+1)|Z^k] - \text{cov}[\tilde{z}(k+1|k)|Z^k] \quad (5.2.2-9)$$

The covariance between the variable to be estimated  $x(k+1)$  and the observation  $z(k+1)$  is

$$P_{xz} \rightarrow \text{cov}[x(k+1), z(k+1)|Z^k] = \text{cov}[\tilde{x}(k+1|k), \tilde{z}(k+1|k)|Z^k] \quad (5.2.2-10)$$

The posterior covariance of the state  $x(k+1)$  — the ***updated state covariance*** — is

$$\begin{aligned} P_{xx|z} \rightarrow P(k+1) &\triangleq P(k+1|k+1) = \text{cov}[x(k+1)|Z^{k+1}] \\ &= \text{cov}[\tilde{x}(k+1|k+1)|Z^{k+1}] \end{aligned} \quad (5.2.2-11)$$

**Note**

Similarly to the single-argument and two-argument notations for the state,  $\bar{x}$  and  $\hat{x}$ , one has the notations  $\bar{P}$  and  $P$  for the one-step predicted and updated state covariances. As before, the two-argument notations are preferred and will be used in the sequel. However, the flowchart of the estimation algorithm will be given with both notations.

**The Filter Gain**

The weighting matrix from the estimation (“updating”) equation (5.2.2-1) becomes the *filter gain*

$$P_{xz} P_{zz}^{-1} \rightarrow W(k+1) \triangleq \text{cov}[x(k+1), z(k+1)|Z^k] S(k+1)^{-1} \quad (5.2.2-12)$$

**Remark**

The reasons for the designation of the above as filter gain are as follows:

1. The recursive estimation algorithm is a filter — it reduces the effect of the various noises on the quantity of interest (the state estimate).
2. The quantity (5.2.2-12) multiplies the observation  $z(k+1)$ , which is the input to the filter; that is, this quantity is a gain.

**5.2.3 Derivation of the Dynamic Estimation Algorithm**

The *predicted state* (5.2.2-4) follows by applying on the state equation (5.2.1-1) the operator of expectation conditioned on  $Z^k$ ,

$$E[x(k+1)|Z^k] = E[F(k)x(k) + G(k)u(k) + v(k)|Z^k] \quad (5.2.3-1)$$

Since the process noise  $v(k)$  is *white and zero mean*, this results in

$$\boxed{\hat{x}(k+1|k) = F(k)\hat{x}(k|k) + G(k)u(k)} \quad (5.2.3-2)$$

Subtracting the above from (5.2.1-1) yields the *state prediction error*

$$\tilde{x}(k+1|k) \triangleq x(k+1) - \hat{x}(k+1|k) = F(k)\tilde{x}(k|k) + v(k) \quad (5.2.3-3)$$

Note the cancellation of the input  $u(k)$  in (5.2.3-3) — it has no effect on the estimation error as long as it is *known*.

The *state prediction covariance* (5.2.2-8) is

$$E[\tilde{x}(k+1|k)\tilde{x}(k+1|k)'|Z^k] = F(k)E[\tilde{x}(k|k)\tilde{x}(k|k)'|Z^k]F(k)' + E[v(k)v(k)'] \quad (5.2.3-4)$$

which can be rewritten as

$$\boxed{P(k+1|k) = F(k)P(k|k)F(k)' + Q(k)} \quad (5.2.3-5)$$

The cross-terms in (5.2.3-4) are zero due to the fact that  $v(k)$  is zero mean and white and, thus, orthogonal to  $\tilde{x}(k|k)$ .

The ***predicted measurement*** (5.2.2-6) follows similarly by taking the expected value of (5.2.1-3) conditioned on  $Z^k$ ,

$$E[z(k+1)|Z^k] = E[H(k+1)x(k+1) + w(k+1)|Z^k] \quad (5.2.3-6)$$

Since the measurement noise  $w(k+1)$  is zero mean and white, this becomes

$$\boxed{\hat{z}(k+1|k) = H(k+1)\hat{x}(k+1|k)} \quad (5.2.3-7)$$

Subtracting the above from (5.2.1-3) yields the ***measurement prediction error***

$$\tilde{z}(k+1|k) \triangleq z(k+1) - \hat{z}(k+1|k) = H(k+1)\bar{x}(k+1|k) + w(k+1) \quad (5.2.3-8)$$

The ***measurement prediction covariance*** in (5.2.2-9) follows from (5.2.3-8), in a manner similar to (5.2.3-5), as

$$\boxed{S(k+1) = H(k+1)P(k+1|k)H(k+1)' + R(k+1)} \quad (5.2.3-9)$$

The covariance (5.2.2-10) between the state and measurement is, using (5.2.3-8),

$$\begin{aligned} E[\tilde{x}(k+1|k)\tilde{z}(k+1|k)'|Z^k] \\ = E[\tilde{x}(k+1|k)[H(k+1)\bar{x}(k+1|k) + w(k+1)]'|Z^k] \\ = P(k+1|k)H(k+1)' \end{aligned} \quad (5.2.3-10)$$

The ***filter gain*** (5.2.2-12) is, using (5.2.3-9) and (5.2.3-10),

$$\boxed{W(k+1) \triangleq P(k+1|k)H(k+1)'S(k+1)^{-1}} \quad (5.2.3-11)$$

Thus the ***updated state estimate*** (5.2.2-7) can be written according to (5.2.2-1) as

$$\boxed{\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + W(k+1)v(k+1)} \quad (5.2.3-12)$$

where

$$\boxed{v(k+1) \triangleq z(k+1) - \hat{z}(k+1|k) = \tilde{z}(k+1|k)} \quad (5.2.3-13)$$

is called the ***innovation*** or ***measurement residual***. This is the same as (5.2.3-8), but the notation  $v$  will be used in the sequel. Note that in view of this,  $S$  is also the ***innovation covariance***.

Finally, the ***updated covariance*** (5.2.2-11) of the state at  $k + 1$  is, according to (5.2.2-2),

$$\begin{aligned} P(k+1|k+1) &= P(k+1|k) - P(k+1|k) \\ &\quad \cdot H(k+1)' S(k+1)^{-1} H(k+1) P(k+1|k) \\ &= [I - W(k+1)H(k+1)]P(k+1|k) \end{aligned} \quad (5.2.3-14)$$

or, in symmetric form,

$$P(k+1|k+1) = P(k+1|k) - W(k+1)S(k+1)W(k+1)' \quad (5.2.3-15)$$

Equation (5.2.3-12) is called the ***state update*** equation, since it yields the updated state estimate, and (5.2.3-15) is the ***covariance update*** equation.

Note the similarity between the state update equation (5.2.3-12) and the recursive LS equation (3.4.2-15). The covariance update equation (5.2.3-15) is analogous to (3.4.2-11). The only difference is that in the LS case the prediction to  $k + 1$  from  $k$  is the same as the updated value at  $k$ . This follows from the fact that in the LS formulation one deals with a constant parameter while in a dynamic system the state evolves in time.

## Alternative Forms for the Covariance Update

Similarly to (3.4.2-6), there is a recursion for the inverse covariance

$$P(k+1|k+1)^{-1} = P(k+1|k)^{-1} + H(k+1)' R(k+1)^{-1} H(k+1) \quad (5.2.3-16)$$

Using the matrix inversion lemma, it can be easily shown that (5.2.3-16) is algebraically equivalent to (5.2.3-15). The filter using (5.2.3-16) instead of (5.2.3-15) is known as the ***information matrix filter*** (see Chapter 7).

As in (3.4.2-12), the filter gain (5.2.3-11) has the alternate expression

$$W(k+1) = P(k+1|k+1)H(k+1)'R(k+1)^{-1} \quad (5.2.3-17)$$

An alternative form for the covariance update equation (5.2.3-15), which holds for an *arbitrary gain*  $W$  (see Problem 5-5), called the ***Joseph form covariance update***, is

$$\begin{aligned} P(k+1|k+1) &= [I - W(k+1)H(k+1)]P(k+1|k) \\ &\quad \cdot [I - W(k+1)H(k+1)]' + W(k+1)R(k+1)W(k+1)' \end{aligned} \quad (5.2.3-18)$$

While this is computationally more expensive than (5.2.3-15), it is less sensitive to round-off errors: It will not lead to negative eigenvalues, as (5.2.3-15) is prone to, due to the subtraction present in it. Numerical techniques that reduce the sensitivity to round-off errors are discussed in Chapter 7.

Equation (5.2.3-18) can be also used for evaluation of the ***sensitivity of the filter to an incorrect gain*** (see Section 5.6). This equation can also be used to obtain the optimal gain (see problem 5-9).

## Intuitive Interpretation of the Gain

Note from (5.2.3-11) that the *optimal filter gain* is (taking a simplistic “scalar view” of it)

1. “Proportional” to the state prediction variance
2. “Inversely proportional” to the innovation variance

Thus, the gain is

- “Large” if the state prediction is “inaccurate” (has a large variance) and the measurement is “accurate” (has a relatively small variance)
- “Small” if the state prediction is “accurate” (has a small variance) and the measurement is “inaccurate” (has a relatively large variance)

A large gain indicates a “rapid” response to the measurement in updating the state, while a small gain yields a slower response to the measurement. In the frequency domain it can be shown that these properties correspond to a higher/lower **bandwidth of the filter**.

A filter whose *optimal gain* is higher yields less “noise reduction,” as one would expect from a filter with a higher bandwidth. This will be seen quantitatively in the next chapter.

## Remark

Equations (5.2.3-9) and (5.2.3-15) yield **filter-calculated covariances**, which are exact if all the modeling assumptions used in the filter derivation hold. In practice this is not always the case and the validity of these filter-calculated estimation accuracies should be tested, as discussed in Section 5.4.

### 5.2.4 Overview of the Kalman Filter Algorithm

Under the *Gaussian assumption* for the initial state (or initial state error) and all the noises entering into the system, the Kalman filter is the **optimal MMSE state estimator**. If these random variables are *not Gaussian* and one has only their first two moments, then, in view of the discussion from Section 3.3, the Kalman filter algorithm is the **best linear state estimator**, that is, the **LMMSE state estimator**.

The flowchart of one cycle of the Kalman filter is presented in Fig. 5.2.4-1 with the two-argument notations and in Fig. 5.2.4-2 with the one-argument notations. Note that at every stage (cycle)  $k$  the entire past is summarized by the *sufficient statistic*  $\hat{x}(k|k)$  and the associated covariance  $P(k|k)$ .

The left-side column represents the true system’s evolution from the state at time  $k$  to the state at time  $k + 1$  with the input  $u(k)$  and the process noise  $v(k)$ . The measurement follows from the new state and the noise  $w(k+1)$ . The

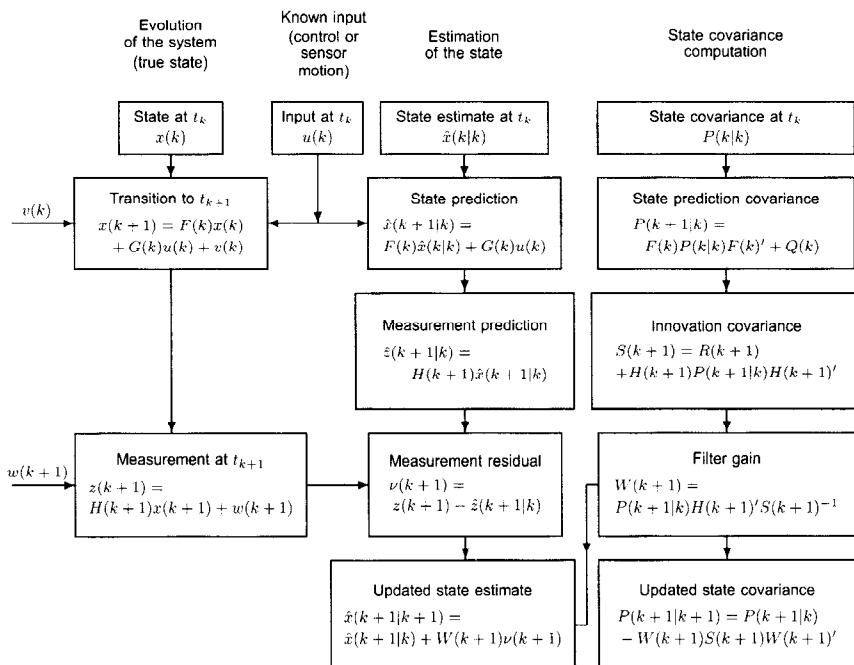
known input (e.g., control, platform motion, or sensor pointing) enters (usually) the system with the knowledge of the latest state estimate and is used by the state estimator to obtain the predicted value for the state at the next time.

The state estimation cycle consists of the following:

1. State and measurement prediction (also called *time update*)
2. State update (also called *measurement update*)

The state update requires the filter gain, obtained in the course of the covariance calculations. The covariance calculations are *independent* of the state and measurements (and control — assumed to be known) and can, therefore, be performed *offline*.

### The Workhorse of Estimation — The Kalman Filter



**Figure 5.2.4-1:** One cycle in the state estimation of a linear system.

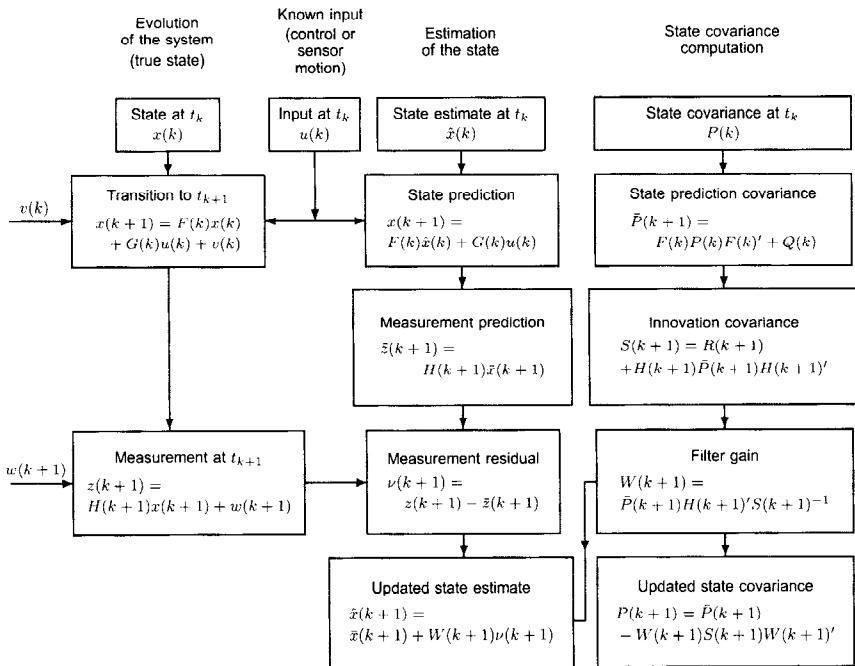


Figure 5.2.4-2: One cycle in the state estimation of a linear system (with single-argument notations).

### Summary of the Statistical Assumptions of the Kalman Filter

The initial state has the known mean and covariance

$$E[x(0)|Z^0] = \hat{x}(0|0) \quad (5.2.4-1)$$

$$\text{cov}[x(0)|Z^0] = P(0|0) \quad (5.2.4-2)$$

where  $Z^0$  denotes the initial (prior) information.

The process and measurement noise sequences are *zero mean and white* with *known covariance matrices*

$$E[v(k)] = 0 \quad (5.2.4-3)$$

$$E[v(k)v(j)'] = Q(k)\delta_{kj} \quad (5.2.4-4)$$

$$E[w(k)] = 0 \quad (5.2.4-5)$$

$$E[w(k)w(j)'] = R(k)\delta_{kj} \quad (5.2.4-6)$$

All the above are *mutually uncorrelated*

$$E[x(0)v(k)'] = 0 \quad \forall k \quad (5.2.4-7)$$

$$E[x(0)w(k)'] = 0 \quad \forall k \quad (5.2.4-8)$$

$$E[v(k)w(j)'] = 0 \quad \forall k, j \quad (5.2.4-9)$$

It can be easily shown that under the Gaussian assumption the whiteness and the uncorrelatedness of the noises imply the following:

$$E[v(k)|Z^k] = E[v(k)] = 0 \quad (5.2.4-10)$$

$$E[w(k)|Z^{k-1}] = E[w(k)] = 0 \quad (5.2.4-11)$$

Property (5.2.4-10) was used in (5.2.3-2), while (5.2.4-11) was used in (5.2.3-7).

### Remark

The dynamic (plant) equation parameters — the matrices  $F$ ,  $G$  — and the measurement equation parameters — the matrix  $H$  — are assumed known.

### Computational Requirements

The *computational requirements* of the KF are approximately proportional to  $n^3$  where  $n = \max(n_x, n_z)$ .

### Some Extensions

The assumptions of

- White process noise
- White measurement noise
- Uncorrelatedness between the process and the measurement noise sequences

can be relaxed.

An autocorrelated (“colored”) noise has to be modeled as the output of a subsystem driven by white noise; that is, it has to be *prewhitened*, as discussed in Chapter 4. For an *autocorrelated process noise*, the state vector has to be augmented to incorporate this subsystem. An example of prewhitening of an autocorrelated process noise is presented in Section 8.2.

The situation where there is *correlation between the two noise sequences* is discussed in Section 8.3. The filter derivation for an *autocorrelated measurement noise*, which can be done without augmenting the state, is presented in Section 8.4.

Discrete-time *smoothing* is presented in Section 8.6.

*Continuous-time state estimation* as an extension of the discrete-time results is discussed in Chapter 9.

### 5.2.5 The Matrix Riccati Equation

As pointed out in Section 3.2, the covariance equations in the static MMSE estimation problem are independent of the measurements. Consequently, the covariance equations for the state estimation problem (in a linear dynamic system), derived in Subsection 5.2.3, can be iterated forward offline.

It can be easily shown that the following recursion can be written for the one-step prediction covariance

$$P(k+1|k) = F(k)\{P(k|k-1) - P(k|k-1)H(k)' \cdot [H(k)P(k|k-1)H(k)' + R(k)]^{-1}H(k)P(k|k-1)\}F(k)' + Q(k) \quad (5.2.5-1)$$

This is the **discrete-time (difference) matrix Riccati equation**, or just the **Riccati equation**. The above follows by substituting (5.2.3-9) and (5.2.3-11) into (5.2.3-15) and substituting the result into (5.2.3-5).

The solution of the above Riccati equation for a time-invariant system converges to a finite **steady-state covariance** if

1. The pair  $\{F, H\}$  is *completely observable*.

If, in addition,

2. The pair  $\{F, C\}$ , where  $Q \triangleq CC'$  ( $C$  is the **Cholesky factor** (see Subsection 7.4.2) — a square root of  $Q$ ), is *completely controllable*,

then the steady-state covariance is a *unique positive definite matrix* — independent of the initial covariance.

The steady-state covariance matrix is the solution of the **algebraic matrix Riccati equation** (or just the **algebraic Riccati equation**)

$$P = F[P - PH'(HPH' + R)^{-1}HP]F' + Q \quad (5.2.5-2)$$

and this yields the **steady-state gain** for the Kalman filter.

The interpretation of the above conditions is as follows:

1. The observability condition on the state guarantees a “steady flow” of information about *each* state component — this prevents the uncertainty from becoming unbounded. This condition yields the existence of a (not necessarily unique) steady-state solution for the covariance matrix that is positive definite or positive semidefinite (i.e., with finite positive or nonnegative eigenvalues, respectively).

2. The controllability condition states that the process noise enters into each state component and prevents the covariance of the state from converging to zero. This condition causes the covariance to be positive definite (i.e., all the eigenvalues are positive).

### Filter Stability

The convergence of the covariance to a *finite* steady state — that is, *the error becoming a stationary process in the MS sense* — is equivalent to **filter**

**stability** in the bounded input bounded output sense.

Stability of the filter does not require the dynamic system to be stable — only the observability condition (1) is required. As indicated above, observability alone does not guarantee uniqueness — the steady-state solution might depend on the initial covariance — but the existence (finiteness) of the solution is the key.

This is particularly important, since the state models used in tracking are *unstable* — they have an integration (from velocity to position). Stability means “bounded input bounded output,” and this condition is not satisfied by an integrator — its continuous-time transfer function has a pole at the origin and in discrete time it has a pole at 1.

## Remarks

If the state covariance matrix is *positive semidefinite* rather than positive definite, that is, it has some zero eigenvalues that reflect the filter’s “belief” that it has “perfectly accurate” estimates of some state components, the gain will be zero for those state components — an *undesirable feature*.

In view of this, in many applications where there is no physical process noise, an *artificial process noise* or *pseudo-noise* is assumed (i.e., a matrix  $Q$  that will lead to condition (2) being satisfied).

## The Riccati Equation and the CRLB

The lower bound on the minimum achievable covariance in state estimation is given by the (posterior) CRLB, which was presented for random parameters in (2.7.2-3). Since the state is a random variable, this is applicable to state estimation.

In the linear Gaussian case it can be shown that this is given by the solution of the Riccati equation. In the more general non-Gaussian case, the solution of the Riccati equation is the covariance matrix (actually matrix MSE) associated with the best linear state estimate; however, as discussed in Subsection 3.3.2, a nonlinear estimator can provide better estimates — with covariance smaller than the solution of the Riccati equation.

## 5.2.6 Properties of the Innovations and the Likelihood Function of the System Model

### The Innovations — a Zero-Mean White Sequence

An important property of the *innovation sequence* is that it is an *orthogonal sequence*, that is,

$$E[\nu(k)\nu(j)'] = S(k)\delta_{kj} \quad (5.2.6-1)$$

where  $\delta_{kj}$  is the Kronecker delta function.

This can be seen as follows. Without loss of generality, let  $j \leq k - 1$ . Use will be made of the smoothing property of the conditional expectations (see Subsection 1.4.12)

$$E[\nu(k)\nu(j)'] = E[E[\nu(k)\nu(j)'|Z^{k-1}]] \quad (5.2.6-2)$$

Note that  $\nu(j)$  is a linear combination of the measurements up to  $j$ ; that is, given  $Z^{k-1}$ , it is *not a random variable anymore* and it can thus be taken outside the inner expectation. This yields

$$E[\nu(k)\nu(j)'] = E[E[\nu(k)|Z^{k-1}]\nu(j)'] \quad (5.2.6-3)$$

The inside expectation in (5.2.6-3) is, in view of (5.2.2-6),

$$E[z(k) - \hat{z}(k|k-1)|Z^{k-1}] = 0 \quad (5.2.6-4)$$

and therefore (5.2.6-1) follows for  $k \neq j$ .

The uncorrelatedness property (5.2.6-1) of the innovations implies that since they are Gaussian, the innovations are independent of each other and thus the innovation sequence is *strictly white*. Without the Gaussian assumption, the innovation sequence is wide sense white.

Thus the innovation sequence is *zero mean and white*.

### Remark

Unlike the innovations, the state estimation errors are not white — they are *correlated in time* (see problem 5-11).

### The Likelihood Function of the System Model

The joint pdf of the measurements up to  $k$ , denoted as

$$Z^k = \{z(j)\}_{j=1}^k \quad (5.2.6-5)$$

can be written as

$$p[Z^k] = p[z(k), Z^{k-1}] = p[z(k)|Z^{k-1}]p[Z^{k-1}] = \prod_{i=1}^k p[z(i)|Z^{i-1}] \quad (5.2.6-6)$$

where  $Z^0$  is the prior information, shown explicitly only in the expression of (5.2.6-6).

If the above pdfs are Gaussian, then

$$\begin{aligned} p[z(i)|Z^{i-1}] &= \mathcal{N}[z(i); \hat{z}(i|i-1), S(i)] = \mathcal{N}[z(i) - \hat{z}(i|i-1); 0, S(i)] \\ &= \mathcal{N}[\nu(i); 0, S(i)] = p[\nu(i)] \end{aligned} \quad (5.2.6-7)$$

Using (5.2.6-7) in (5.2.6-6) yields

$$p[Z^k] = \prod_{i=1}^k p[\nu(i)] \quad (5.2.6-8)$$

that is, the joint pdf of the sequence of measurements  $Z^k$  is equal to the product of the marginal pdfs of the corresponding innovations. This shows the *informational equivalence of the measurements and the innovations*.

Since (5.2.6-8) is the joint pdf of  $Z^k$  conditioned on the system model (not indicated explicitly), it is the *likelihood function of the system model*. This will be used in Chapter 11 to evaluate the “goodness” of models in multiple model adaptive filtering.

### 5.2.7 The Innovations Representation

The counterpart of the Riccati equation that yields the recursion of the one-step prediction covariance  $P(k+1|k)$  is the recursion of the one-step prediction of the state  $\hat{x}(k+1|k)$ , called the *innovations representation*.

This is obtained from (5.2.3-2) and (5.2.3-12), without the deterministic input, for simplicity, as

$$\begin{aligned} \hat{x}(k+1|k) &= F(k)\hat{x}(k|k-1) + F(k)W(k)\nu(k) \\ &= F(k)\hat{x}(k|k-1) + W_i(k)[z(k) - H(k)\hat{x}(k|k-1)] \end{aligned} \quad (5.2.7-1)$$

where

$$W_i(k) \triangleq F(k)W(k) \quad (5.2.7-2)$$

is the gain in the innovations representation (sometimes called ambiguously the filter gain).

Equation (5.2.7-1) can also be rewritten as the state equation

$$\hat{x}(k+1|k) = [F(k) - W_i(k)H(k)]\hat{x}(k|k-1) + W_i(k)z(k) \quad (5.2.7-3)$$

with the input being the (nonwhite) sequence  $z(k)$  and the output being the innovation

$$\nu(k) = -H(k)\hat{x}(k|k-1) + z(k) \quad (5.2.7-4)$$

This motivates the name innovations representation for the system (5.2.7-3) and (5.2.7-4).

## The Kalman Filter as a Whitening System

Note that while the input to this system is not white, its output is a white sequence. Thus the state estimation filter, written as (5.2.7-3) and (5.2.7-4), can be seen as a *whitening system for the measurement sequence*.

### 5.2.8 Some Orthogonality Properties

In the static estimation problem the LMMSE estimator was derived based on the principle of orthogonality, which states that the estimation error  $\tilde{x}$  has to be orthogonal to the observation(s)  $z$ , that is,

$$\tilde{x} \perp z \iff \langle \tilde{x}, z \rangle \triangleq E[\tilde{x}z] = 0 \quad (5.2.8-1)$$

The state estimator for linear dynamic systems — the Kalman filter — while derived under the Gaussian assumption, is (as pointed out earlier) the LMMSE estimator. Therefore, the orthogonality properties carry over.

Note that the estimate is a linear function of the measurements

$$\hat{x}(k|k) = L_k(Z^k) \quad \forall k \quad (5.2.8-2)$$

or, in a more general manner,

$$\hat{x}(i|k) = L_{i,k}(Z^k) \quad \forall i, k \quad (5.2.8-3)$$

where  $L$  denotes a *linear transformation* (because the estimates are linear functions of the measurements) and the measurement set  $Z^k$  includes the initial information  $Z^0$ .

Thus, the estimation error

$$\tilde{x}(i|k) \triangleq x(i) - \hat{x}(i|k) \quad (5.2.8-4)$$

has the following *orthogonality properties*

$$\tilde{x}(i|k) \perp z(j) \quad \forall j \leq k \quad (5.2.8-5)$$

$$\tilde{x}(i|k) \perp \hat{x}(l|j) \quad \forall j \leq k, \quad \forall i, l \quad (5.2.8-6)$$

With the Gaussian assumption, all the orthogonality properties — which are equivalent to uncorrelatedness — also imply independence.

### 5.2.9 The Kalman Filter — Summary

The *MMSE state estimation model* for a dynamic system consists of the following:

- *Initial state* — unknown, assumed to be a *random variable* with a certain mean (initial estimate) and covariance (measure of the accuracy of the initial estimate).
- Evolution of the *system's state* — according to a possibly time-varying *linear difference equation* (*plant equation* or *dynamics*) driven by
  - A known input (the control);
  - An additive random disturbance — the *process noise* — a zero-mean white (uncorrelated) stochastic process with a known, possibly time-varying, covariance.
- *Measurements* — a *linear function* of the state with an additive random disturbance (*measurement noise*), which is a zero-mean white stochastic process with a known, possibly time-varying, covariance.

If, in addition, all the random variables of the problem, namely,

- The initial state,
- The process noises, and
- The measurement noises

are *Gaussian and mutually independent* (i.e., under the *LG assumption*), the MMSE estimate of the state of the system under consideration — the conditional mean of the state given the measurements — is given by the (discrete-time) Kalman filter.

The discrete-time Kalman filter computes recursively the MMSE estimate of the state of a dynamic system through the following stages:

- Starting from the *current updated state estimate* (estimate of the current state given the observations up to and including the current time), the *predicted value of the state* for the next sampling time is computed.
- Using the predicted state, the *predicted value* of the next *measurement* is calculated.
- When the new measurement is obtained, the difference between it and its predicted value — the *innovation* (residual) — is evaluated.
- The *updated state at the next time* is obtained as the sum of the predicted state and the correction term. The correction term is the product of the *filter gain* (obtained separately) and the innovation.

Covariance and filter gain calculation:

- Starting from the current *updated state covariance*, the *state prediction covariance* is computed.
- Using the state prediction covariance, the *measurement prediction covariance* (which is the same as the *innovation covariance*) is obtained.
- The *filter gain* is calculated from the state and measurement prediction covariances.
- The *updated covariance* associated with the next state is then computed.

The covariance prediction and update equations combined together result in the (*discrete time*) *matrix Riccati equation*.

For a *time-invariant system*, if it is *observable*, the state estimation covariance will be finite and the Riccati equation will converge to a steady-state solution. If, in addition, the process noise excites each state component, then the steady-state covariance is also positive definite and unique.

The gain of the filter reflects the relative accuracy of the predicted state versus the new measurement:

- If the new measurement is deemed “more accurate” than the predicted state, then the filter gain will be relatively high.
- If the predicted state is deemed “more accurate” than the new measurement, then the gain will be low.

If the state covariance matrix is positive semidefinite, this reflects the filter’s “belief” that it has “perfectly accurate” estimates of some state components — an undesirable feature.

The innovation sequence is *zero mean, white (uncorrelated)*, with covariance equal to the measurement prediction covariance.

The joint pdf of the sequence of measurements, which is the *likelihood function of the model*, is equal to the product of the marginal pdfs of the corresponding innovations, which are Gaussian under the LG assumption.

If the random variables in the state estimation problem for a linear system are *not Gaussian* and one only has their first two moments, then the Kalman filter is the *LMMSE estimator* and the covariances are really the corresponding MSE matrices. In this case, the likelihood function (needed in an adaptive multiple model estimation) is, typically, still assumed to be Gaussian, even though this is clearly an approximation.

## Remark

One can summarize in a nutshell what the KF does: It estimates/predicts the state of a linear system with *zero-mean white noise* disturbances — it reduces the uncertainties due to such disturbances. While this seems very restrictive, through a number of extensions, this technique can be used in numerous problems where the disturbances do not satisfy the above conditions. The essence of the extensions is to map the problem at hand into one that meets (or nearly meets) the above constraints.

### 5.3 EXAMPLE OF A FILTER

#### 5.3.1 The Model

Consider the system with state

$$\dot{x} = \begin{bmatrix} \xi \\ \dot{\xi} \end{bmatrix} \quad (5.3.1-1)$$

which evolves according to

$$x(k+1) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} v(k) \quad k = 0, 1, \dots, 99 \quad (5.3.1-2)$$

with initial condition

$$x(0) = \begin{bmatrix} 0 \\ 10 \end{bmatrix} \quad (5.3.1-3)$$

This represents a one-dimensional motion with position  $\xi$  and velocity  $\dot{\xi}$  sampled at intervals  $T$ , which will be assumed as unity in the sequel.

Note that (5.3.1-2) is of the form

$$x(k+1) = Fx(k) + \Gamma v(k) \quad (5.3.1-4)$$

The process noise, a scalar, which models the acceleration, is a zero-mean white sequence with variance

$$E[v(k)^2] = q \quad (5.3.1-5)$$

The measurements consist of the state's position component corrupted by additive noise

$$z(k) = [1 \ 0] x(k) + w(k) \quad k = 1, \dots, 100 \quad (5.3.1-6)$$

where the measurement noise is a zero-mean white sequence with variance

$$E[w(k)^2] = r = 1 \quad (5.3.1-7)$$

The two noise sequences are mutually independent.

The filter was initialized according to the “two-point differencing” procedure discussed later in Section 5.5 in (5.5.3-3) to (5.5.3-5) with the first two measurements at  $k = 1, 2$  and the filter started running at  $k = 3$ .

See Subsection 5.3.3 for a step-by-step demonstration of using DynaEst<sup>TM</sup> to set up a simulation scenario and design an estimator.

### 5.3.2 Results for a Kalman Filter

Using DynaEst, Figs. 5.3.2-1 through 5.3.2-3 present

1. The true and estimated trajectories of the system in the state space (position-velocity)
2. The variances of the predicted position,  $P_{11}(k|k-1)$ , and updated position,  $P_{11}(k|k)$
3. The variances of the predicted velocity,  $P_{22}(k|k-1)$ , and updated velocity,  $P_{22}(k|k)$
4. Filter gains

for three cases with different values of the process noise variance  $q$ .

In the first case, with  $q = 0$ , the controllability condition 2 of Subsection 5.2.5 does not hold and the state estimation covariance matrix is seen from Fig. 5.3.2-1b to converge to zero. In this case the filter gain also converges to zero as shown in Fig. 5.3.2-1c. The reason this is an undesirable feature is that the filter's "belief" of having a *perfect state estimate* (which "shuts it off") hinges on the *assumed perfect noiseless constant velocity motion*. In practice such an assumption usually does not hold, except for short periods of time.

The filter for  $q = 0$  is equivalent to the LS estimation of the initial position and velocity (straight line fitting), as discussed in Subsection 3.5.1.

For the nonzero values of  $q$ , the filter is seen from Figs. 5.3.2-2 and 5.3.2-3 to reach steady state quite rapidly, in a few time steps.

In Fig. 5.3.2-2, which shows the results on a motion with  $q = 1$ , the steady-state filter gain is  $[0.75 \ 0.50]'$ . In Fig. 5.3.2-3, which corresponds to a strong process noise ( $q = 9$ ) that models a "highly unpredictable" motion, the (optimal) steady-state gain is  $[0.90 \ 0.94]'$  (i.e., higher). Note how the case with higher gain (due to the higher process noise level) leads to a larger state variance — less accurate state estimates.

### 5.3.3 A Step-by-Step Demonstration of DynaEst<sup>TM</sup>

The companion software *DynaEst*<sup>TM</sup>, written in MATLAB<sup>TM</sup>, can be used to set up a simulation scenario, generate measurements, and design an estimator to obtain state estimates. Alternatively, one can import measurements from an external file — for example, obtained using a real sensor — and run the tracker on those real measurements. DynaEst can also be used to analyze the performance of the estimator in terms of RMS estimation errors and filter consistency. In the following, a step-by-step demonstration of designing the simulated scenario in Section 5.3.1 and the Kalman filter in 5.3.2 using DynaEst is presented.

**Starting DynaEst** Start MATLAB as you would normally and, from the MATLAB prompt, change to the directory where DynaEst files are stored. For example, if DynaEst files are stored in directory /matlab/dynaest,

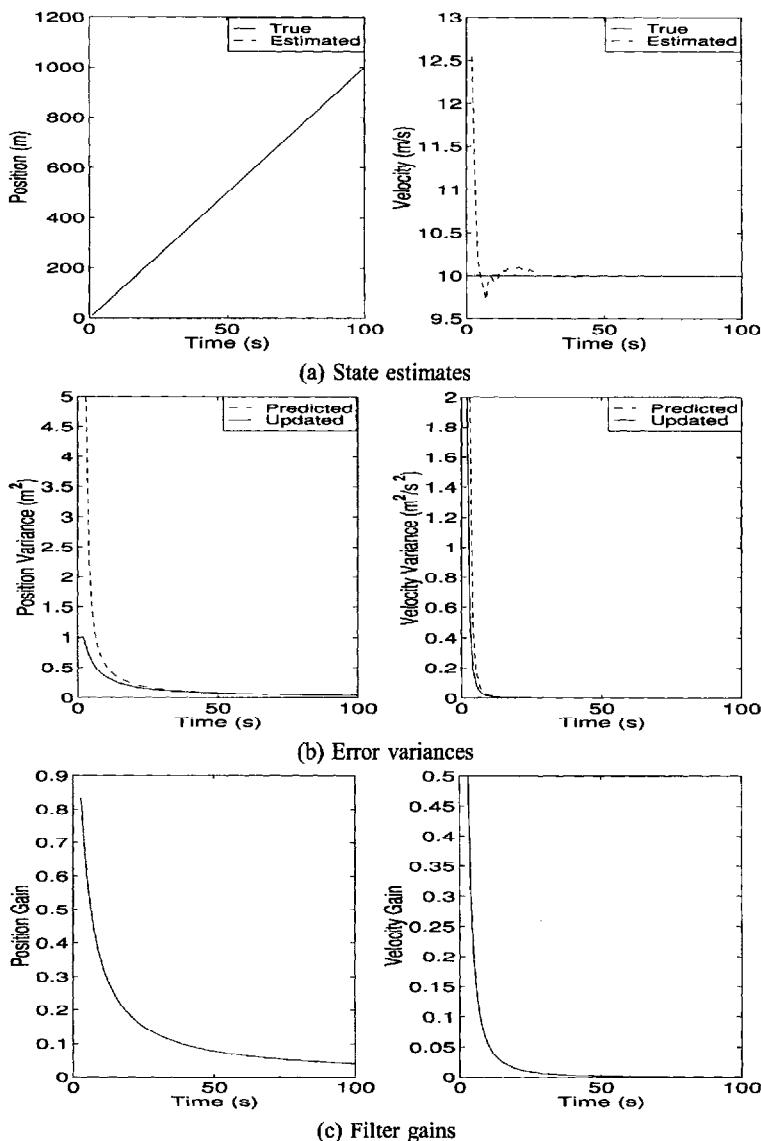
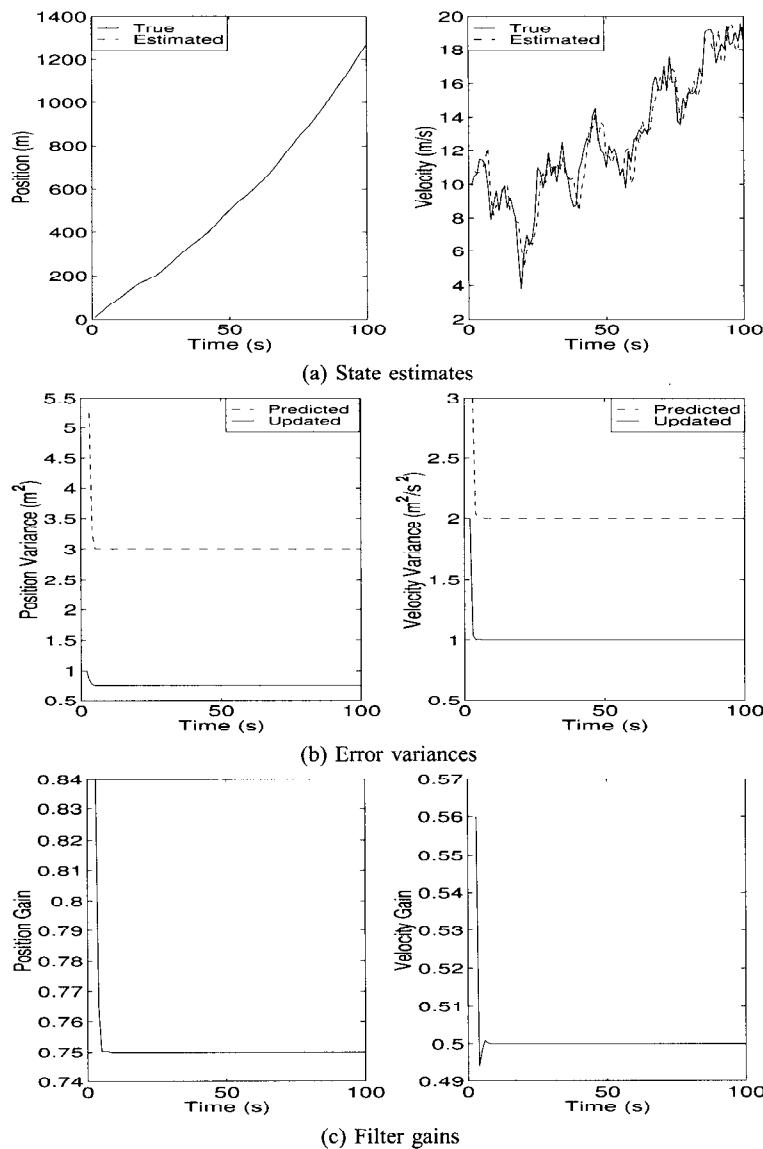
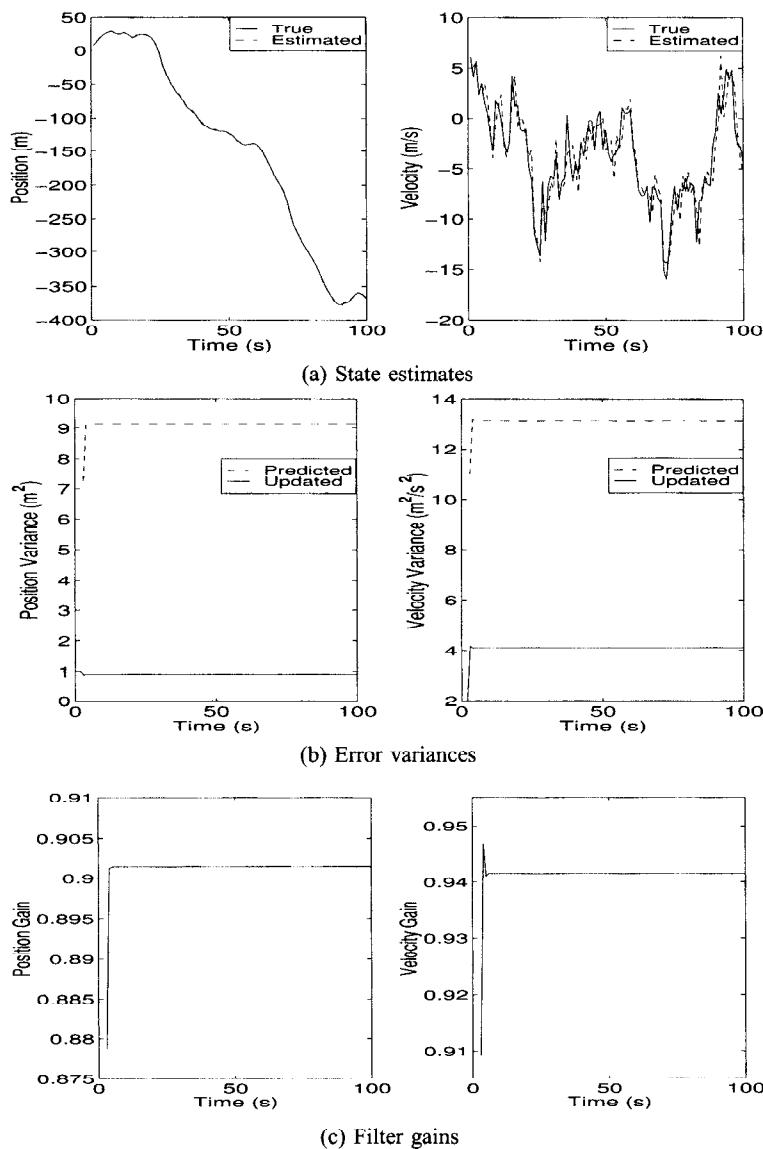


Figure 5.3.2-1: State estimates, error variances and filter gains for  $q = 0$ .



**Figure 5.3.2-2:** State estimates, error variances and filter gains for  $q = 1$ .



**Figure 5.3.2-3:** State estimates, error variances and filter gains for  $q = 9$ .



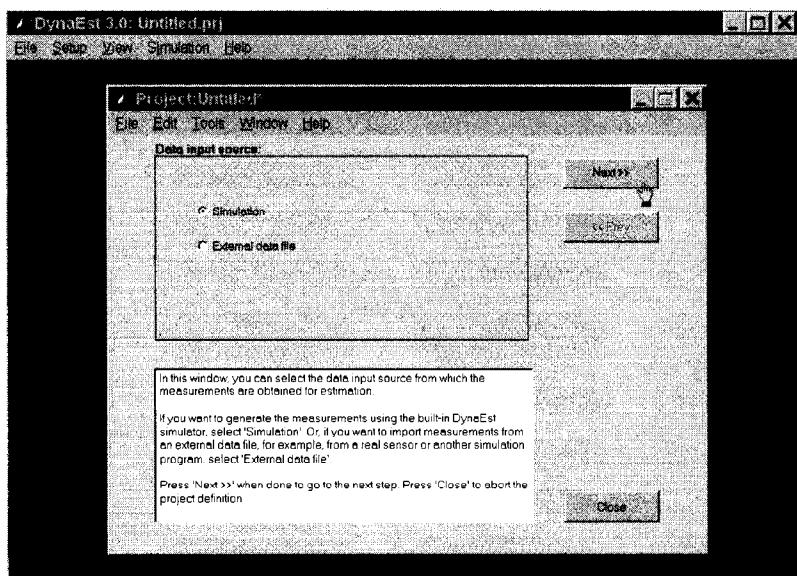
**Figure 5.3.3-1:** Creating a new project in DynaEst.

type `cd /matlab/dynaest`. And then DynaEst can be started by typing `dynaest` at the prompt.

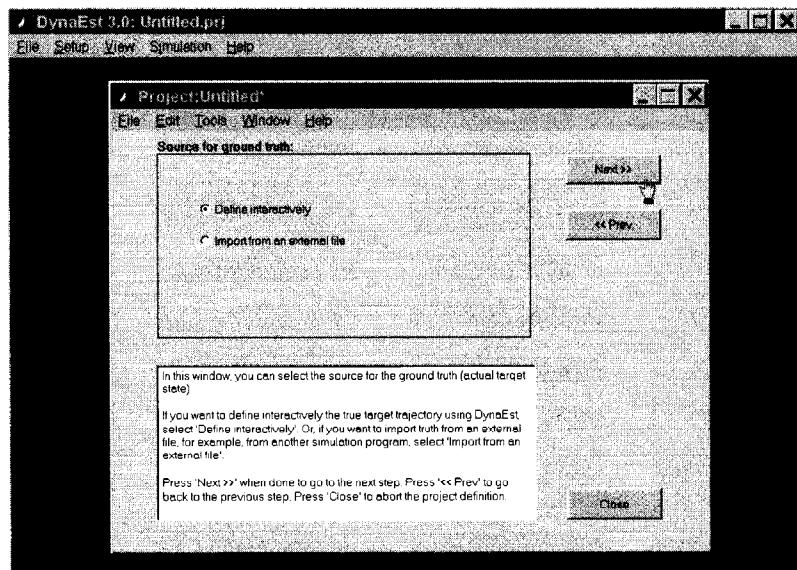
**Creating a Project** In DynaEst, the simulation and estimation parameters are stored as a project in a file with extension `prj`. To create a new project, select the `New` option in the menu `File` as shown in Fig. 5.3.3-1. After specifying the parameters for simulation and estimation, the project can be saved in a file by issuing the `Save As` command in the `File` menu. The project file name is to be given following the command `Save As`. Alternatively, one can open an already saved project using the `Open` option under the same menu.

**Data Source** When the option to create a new project is selected, one can specify the data source for estimation. As mentioned earlier, one can generate **simulated measurements** or use **imported measurements** from a real sensor for estimation. To simulate the scenario in Subsection 5.3.1, select `Simulation` and press `Next` as shown in Fig. 5.3.3-2.

**Ground Truth** When the option to simulate the system is specified, one can also set the system dynamics (defined in the state space) interactively within DynaEst or from an external file. To simulate the scenario in Subsection 5.3.1, select `Define` interactively and press `Next` as shown in Fig. 5.3.3-3.



**Figure 5.3.3-2:** Selecting the data source for estimation.



**Figure 5.3.3-3:** Specifying the ground truth.

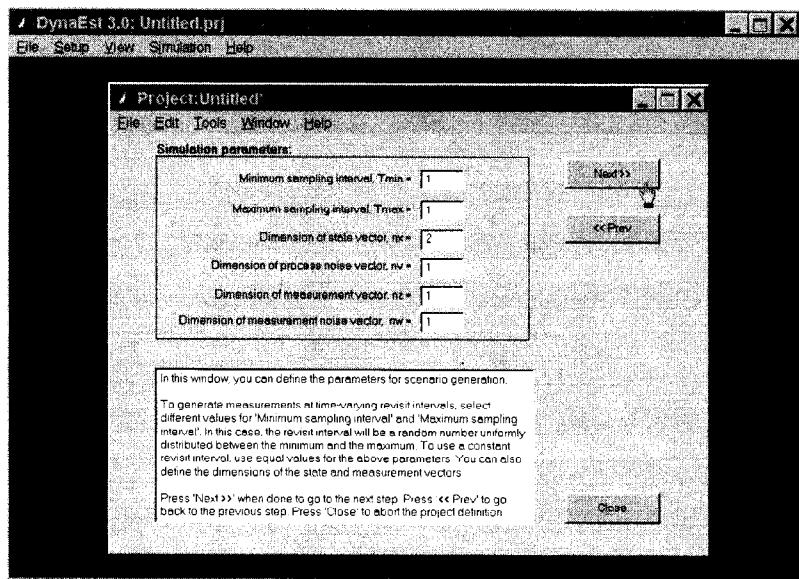


Figure 5.3.3-4: Specifying the system parameters.

**System Parameters** To simulate the scenario in Subsection 5.3.1, specify the parameters of the system defined in (5.3.1-1)–(5.3.1-7) and press Next as shown in Fig. 5.3.3-4. In this step, the dimensions of the state and noise vectors and system matrices are set up.

**Simulation Parameters** Since the measurements are obtained using simulation, one has to specify the number of Monte Carlo runs used for measurement generation and estimation. In addition, the total duration (number of sampling times or *revisits*) of simulation in each run has to be specified as well.

With DynaEst it is possible to generate true target trajectories consisting of a number of motion legs — for example, a constant velocity motion leg followed by a coordinated turn or linear acceleration/deceleration. Since the scenario in Subsection 5.3.1 consists of only a straight line segment, the number of legs is set to one. Specify the parameters and press Next as shown in Fig. 5.3.3-5.

**Motion Equations** Then the true system matrix  $F(k)$  and the true noise gain matrix  $\Gamma$  in the true target state evolution, defined by (5.3.1-3), are specified. DynaEst has the capability to parse matrices given as functions of the sampling interval  $T$ . Define the matrices and press Next as shown in Figs. 5.3.3-6 and 5.3.3-7.

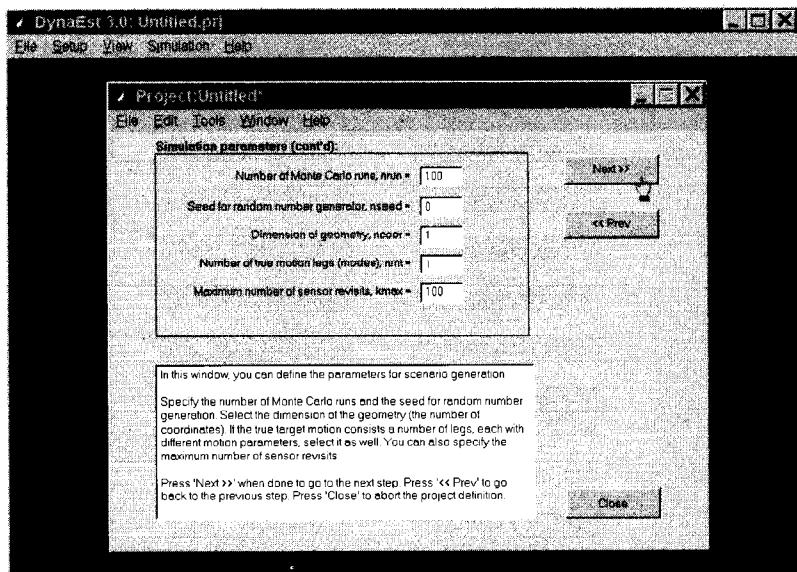


Figure 5.3.3-5: Specifying the simulation parameters.

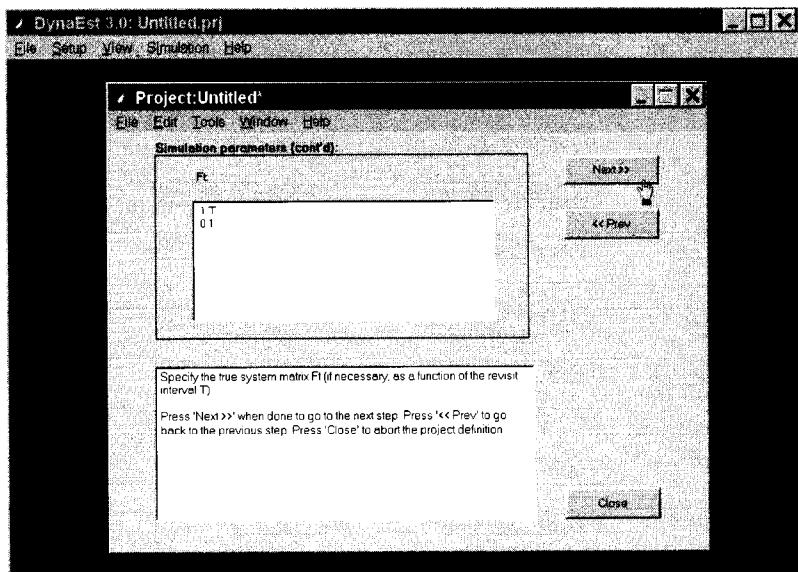
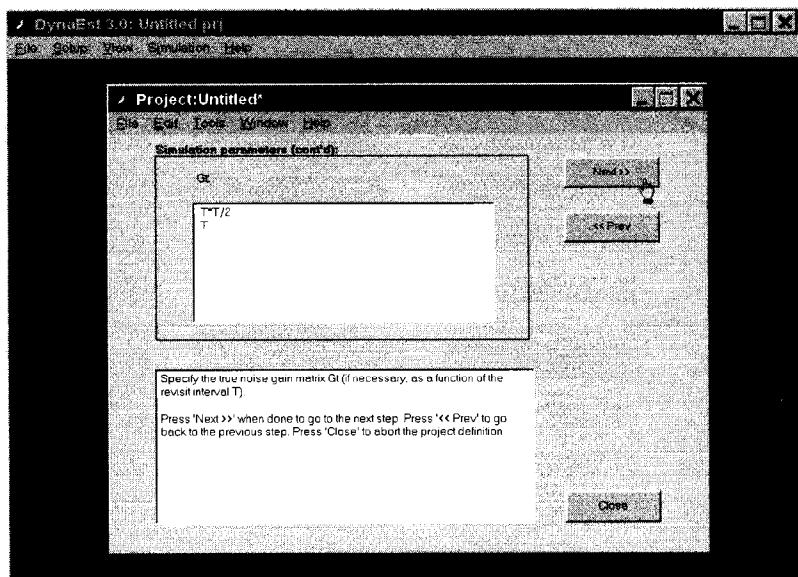


Figure 5.3.3-6: Specifying the system matrix.



**Figure 5.3.3-7:** Specifying the noise gain matrix.

**Measurement Equation** The measurement equation (5.3.1-6) is specified as shown in Fig. 5.3.3-8.

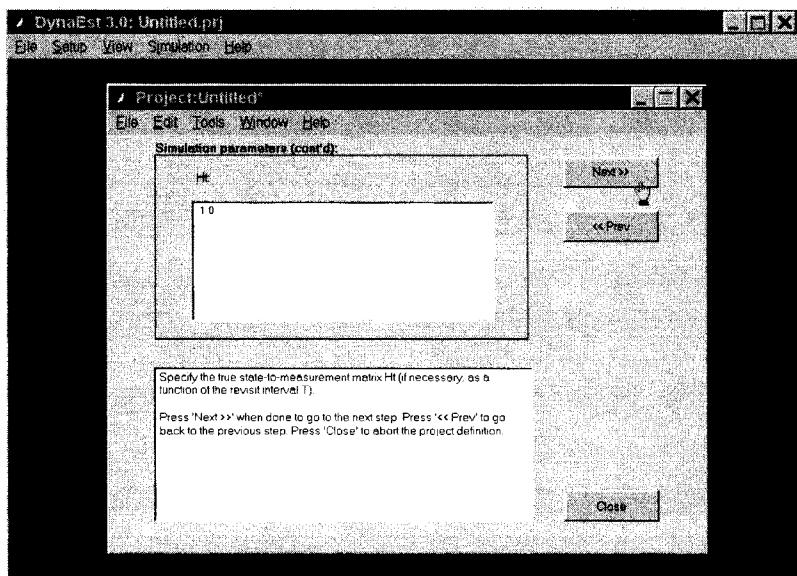
**Noise Parameters** The process and measurement noise variances are specified as shown in Figs. 5.3.3-9 and 5.3.3-10, respectively. Note that  $q = 1$  in Fig. 5.3.3-10.

**Initial Conditions** Finally, the initial state conditions are specified as part of the scenario definition as shown in Fig. 5.3.3-11.

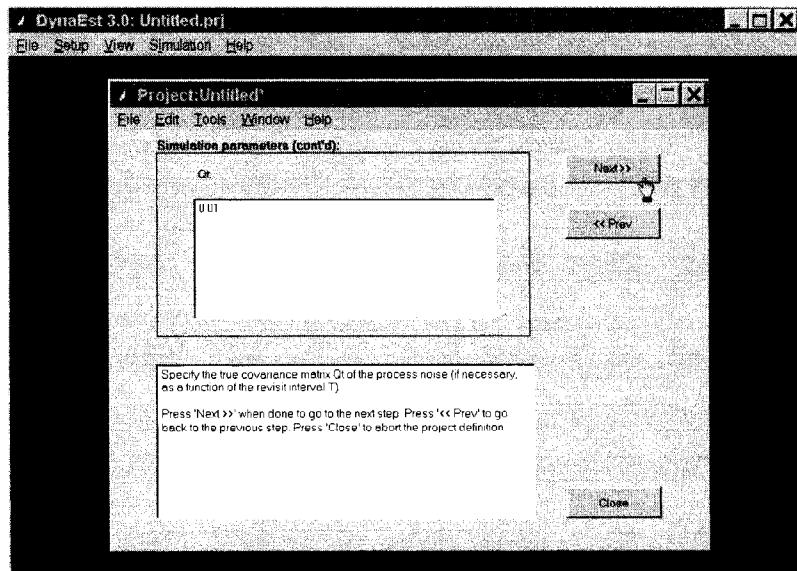
**Using the Defined Scenario** With the above steps, the definition of the simulation scenario is complete and the scenario can be used for estimation, exporting measurements or for sensitivity analysis. For tracking, select the option as in Fig. 5.3.3-12 and proceed to the next step.

**Selecting a Tracker** DynaEst offers a variety of linear estimators, ranging from the simple  $\alpha\text{-}\beta$  filter to the sophisticated IMM estimator. In this section, the design of a Kalman filter is demonstrated. Select the Kalman filter as the choice for estimation as shown in Fig. 5.3.3-13.

**Selecting the Motion Model in the Estimator** The crucial step in the estimator design is the selection of motion model and the noise parameters. In practice, one does not know the true values and has to make a good “engineer’s” choice for these design parameters. DynaEst offers a sequence of



**Figure 5.3.3-8:** Specifying the state-to-measurement matrix.



**Figure 5.3.3-9:** Specifying the process noise variance.

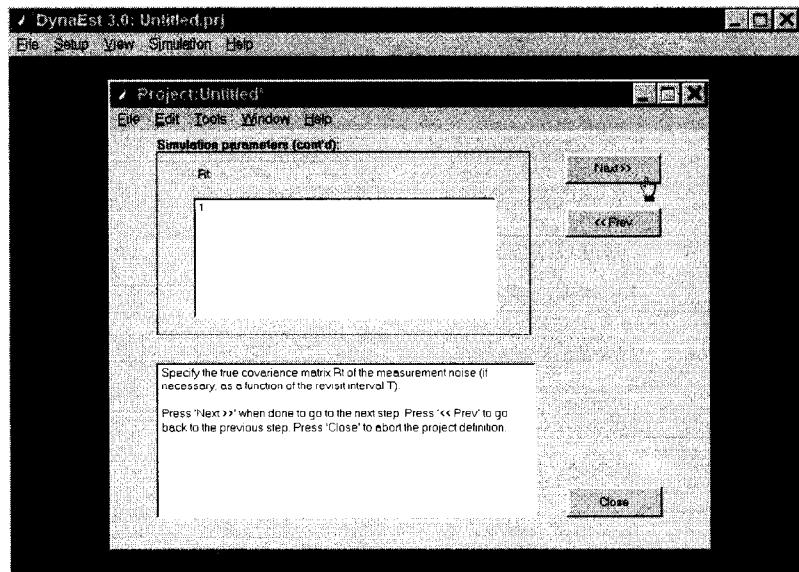


Figure 5.3.3-10: Specifying the measurement noise variance.

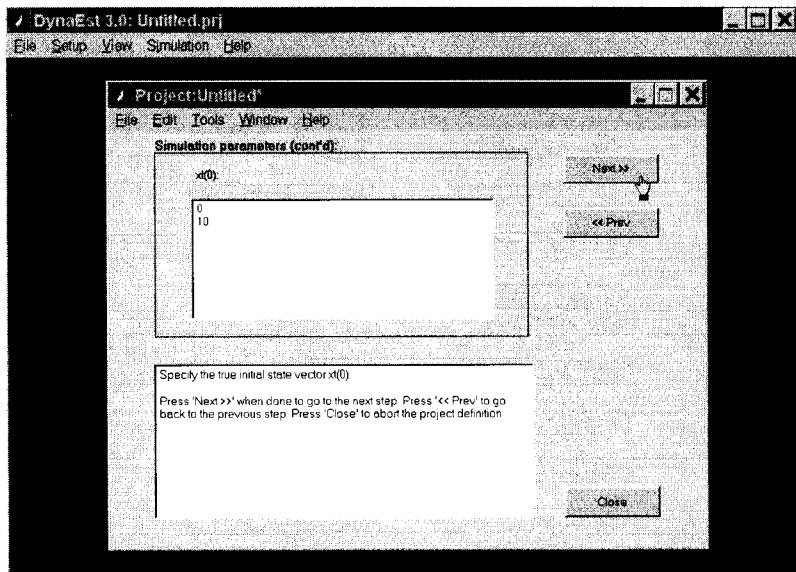


Figure 5.3.3-11: Specifying the initial conditions.

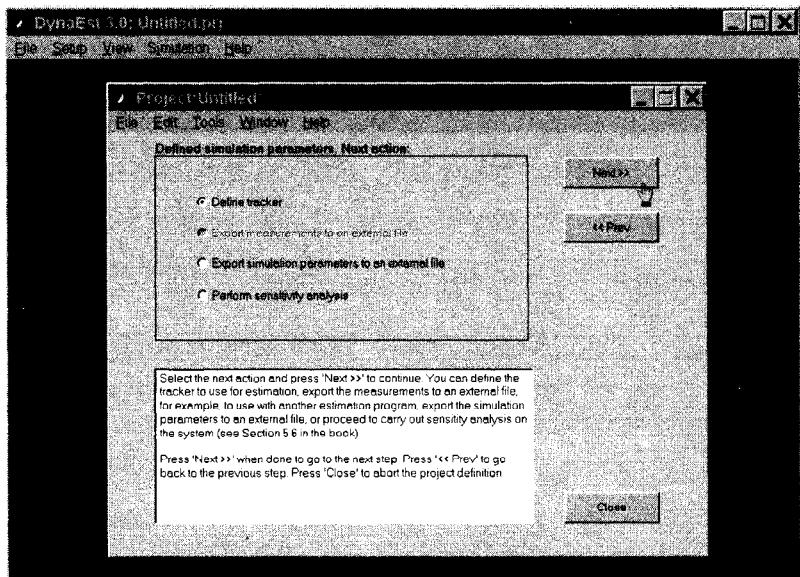


Figure 5.3.3-12: Using the defined scenario.

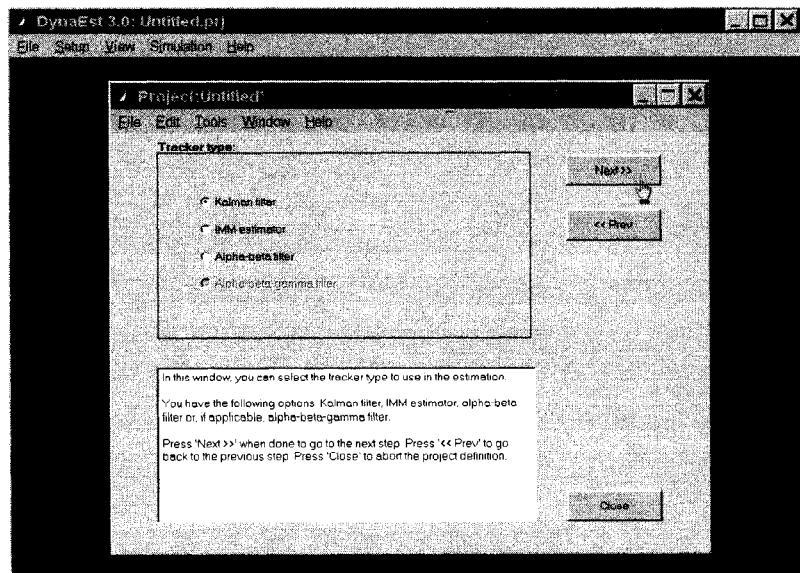
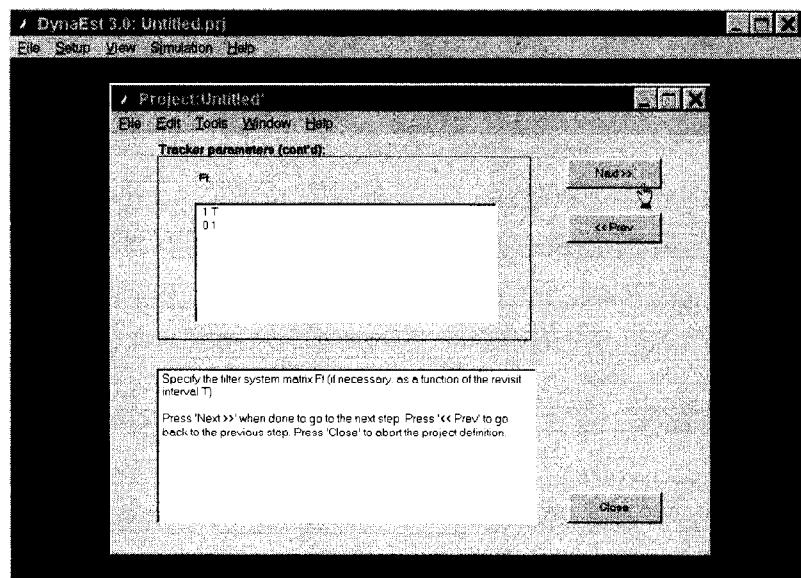


Figure 5.3.3-13: Selecting a tracker.



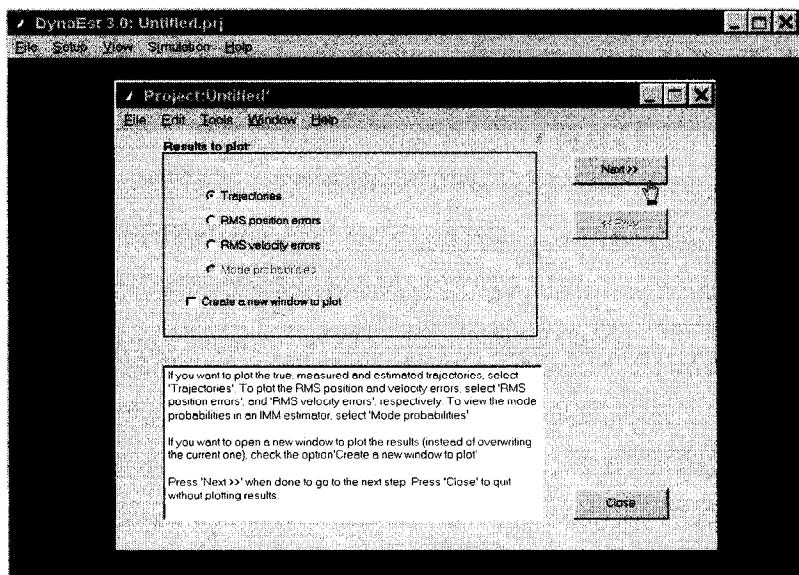
**Figure 5.3.3-14:** Designing the tracker.

windows similar to those in Figs. 5.3.3-6–5.3.3-11 for setting the motion model, noise parameters, initial estimates, and the corresponding covariance matrix. The first window in the estimator design process is shown in Fig. 5.3.3-14.

**Viewing Estimation Results** After the filter design parameters are specified, DynaEst executes the specified number of Monte Carlo runs and offers a choice to plot the estimation results as shown in Fig. 5.3.3-15.

Select the desired performance metric to view and obtain plots similar to those in Fig. 5.3.2-1–5.3.2-3.

The objective of DynaEst is to provide a user-friendly way to design and test linear estimators. However, it also provides the source code in MATLAB so that the user can modify the code to improve the built-in estimators or add additional ones. DynaEst contains far more functionalities than those demonstrated in this simple walk-through. The rest are left as an exercise to the reader.



**Figure 5.3.3-15:** Viewing estimation results.

## 5.4 CONSISTENCY OF STATE ESTIMATORS

### 5.4.1 The Problem of Filter Consistency

In the problem of estimating a parameter that is constant, consistency of an estimator (i.e., a *static estimator*) was defined as *convergence of the estimate to the true value*. This implies that there is a steadily increasing amount of information (in the sense of Fisher) about the parameter that asymptotically reduces to zero the uncertainty about its true value.

When estimating the state of a dynamic system, in general, no convergence of its estimate occurs. What one has, in addition to the “current” estimate of the state,  $\hat{x}(k|k)$ , is the associated covariance matrix,  $P(k|k)$ .

The phenomenon of *divergence* has been observed: Sometimes the filter yields unacceptably large state estimation errors.<sup>2</sup> These can be due to one or more of the following:

- Modeling errors
- Numerical errors
- Programming errors

<sup>2</sup>In the 1960s, when one of the first Kalman filters performed poorly in an avionics application, it was called “the worst invention of the decade.” At that time the design of state estimators was still in the domain of “black magic.”

The question of what an **acceptable estimation error** is, will be discussed in the sequel.

Under the *Linear-Gaussian (LG)* assumption, the conditional pdf of the state  $x(k)$  at time  $k$  is

$$p[x(k)|Z^k] = \mathcal{N}[x(k); \hat{x}(k|k), P(k|k)] \quad (5.4.1-1)$$

The modeling of the system consists of the dynamic equation, the measurement equation, and the statistical properties of the random variables entering into these equations. If all these are completely accurate, then (5.4.1-1) holds exactly. Since all models contain some approximations in practice, it is of interest to what extent one can verify (5.4.1-1).

### Practical Evaluation of Consistency

The **statistical characterization of the disturbances** is usually done with moments up to second order and the resulting filter will then (hopefully) give approximate first and second order moments of the state.

In view of this, (5.4.1-1) is replaced by the two moment conditions

$$E[x(k) - \hat{x}(k|k)] \triangleq E[\tilde{x}(k|k)] = 0 \quad (5.4.1-2)$$

$E[(x(k) - \hat{x}(k|k))(x(k) - \hat{x}(k|k))'] \triangleq E[\tilde{x}(k|k)\tilde{x}(k|k)'] = P(k|k) \quad (5.4.1-3)$

that the filter should satisfy in spite of its inherent approximations.

Condition (5.4.1-2) is the **unbiasedness** requirement for the estimates (i.e., zero-mean estimation error), while (5.4.1-3) is the **covariance matching** requirement — that is, that the **actual MSE** (left-hand side) matches the **filter-calculated covariance** (right-hand side).

Note that if there is a bias, this will increase the MSE, which is the bias squared plus the variance in the scalar case. Thus the test to be discussed in the next subsection will deal with the MSE. This test will be based on the fact that, under the LG assumption, one has

$$E[\tilde{x}(k|k)'P(k|k)^{-1}\tilde{x}(k|k)] = n_x \quad (5.4.1-4)$$

that is, the average of the **squared norm of the estimation error** indicated above has to be equal to the dimension of the corresponding vector since it is chi-square distributed (see Subsection 1.4.17).

### Consistency and Optimality

Since the filter gain is based on the filter-calculated error covariances, it follows that **consistency is necessary for filter optimality**: Wrong covariances yield wrong gain.

This is why **consistency evaluation** is vital for verifying a filter design — it amounts to **evaluation of estimator optimality**.

### 5.4.2 Definition and the Statistical Tests for Filter Consistency

A state estimator (filter) is called **consistent** if its state estimation errors satisfy (5.4.1-2) and (5.4.1-3). This is a **finite-sample consistency** property, that is, the estimation errors based on a finite number of samples (measurements) should be consistent with their theoretical statistical properties:

1. Have mean zero (i.e., the estimates are unbiased).
2. Have covariance matrix as calculated by the filter.

In contradistinction, the parameter estimator consistency is an asymptotic (infinite size sample) property.

The **consistency criteria of a filter** are as follows:

- (a) The state errors should be acceptable as zero mean and have magnitude commensurate with the state covariance as yielded by the filter.
- (b) The innovations should also have the same property.
- (c) The innovations should be acceptable as white.

The last two criteria are the only ones that can be tested in applications with *real data*. The first criterion, which is really the most important one, can be tested only in simulations.

Using the notation

$$\tilde{x}(k|k) = x(k) - \hat{x}(k|k) \quad (5.4.2-1)$$

define the **normalized (state) estimation error squared (NEES)** as

$$\boxed{\epsilon(k) = \tilde{x}(k|k)' P(k|k)^{-1} \tilde{x}(k|k)} \quad (5.4.2-2)$$

The test to be presented next is based on the above quadratic form, and it can verify simultaneously both properties 1 and 2.

Under hypothesis  $H_0$  that the filter is consistent and the LG assumption,  $\epsilon(k)$  is chi-square distributed with  $n_x$  degrees of freedom, where  $n_x$  is the dimension of  $x$ . Then

$$E[\epsilon(k)] = n_x \quad (5.4.2-3)$$

and the test is whether (5.4.2-3) can be accepted.

#### Monte Carlo Simulation Based Tests

The test will be based on the results of **Monte Carlo simulations (runs)** that provide  $N$  independent samples  $\epsilon^i(k)$ ,  $i = 1, \dots, N$ , of the random variable  $\epsilon(k)$ . Let the sample average of  $\epsilon(k)$  — the ( $N$ -run) **average NEES** — be

$$\bar{\epsilon}(k) = \frac{1}{N} \sum_{i=1}^N \epsilon^i(k) \quad (5.4.2-4)$$

Then  $N\bar{\epsilon}(k)$  will have, under  $H_0$ , a chi-square density with  $Nn_x$  degrees of freedom.

Hypothesis (5.4.2-3), that the *state estimation errors are consistent with the filter-calculated covariances* — criterion (a), also called the **chi-square test** — is accepted if

$$\bar{\epsilon}(k) \in [r_1, r_2] \quad (5.4.2-5)$$

where the **acceptance interval** is determined such that

$$P\{\bar{\epsilon}(k) \in [r_1, r_2] | H_0\} = 1 - \alpha \quad (5.4.2-6)$$

For example, with  $\alpha = 0.05$ ,  $n_x = 2$ , and  $N = 50$ , one has from (1.5.4-6), for a two-sided interval,  $r_1 = 1.5$  and  $r_2 = 2.6$ . The interval given in (5.4.2-5) is then the (two-sided) 95% **probability concentration region** for  $\bar{\epsilon}(k)$ .

If  $N = 1$  (i.e., a single run), one can also use this test. The two-sided 95% interval in this case is  $[0.05, 7.38]$ . Note the much narrower range of the interval corresponding to  $N = 50$  Monte Carlo runs — this illustrates the **variability reduction** in such repeated simulations.

Note that a bias in the state estimation error will increase (5.4.2-2), and, if significant, it will yield unacceptably large values for the statistic (5.4.2-4).

If (5.4.2-5) is not satisfied, then a separate bias test using the sample mean of (5.4.2-1) should be carried out to identify the source of the problem. This can be done by taking each component of the state error, divided by its standard deviation, which makes it (under ideal conditions)  $\mathcal{N}(0, 1)$ , and testing to see if its mean can be accepted as zero.

The test statistic for the **normalized mean estimation error (NMEE)** for component  $j$  of the state from runs  $i = 1, \dots, N$  is

$$\mu_j(k) = \frac{1}{N} \sum_{i=1}^N \frac{\tilde{x}_j^i(k|k)}{\sqrt{P_{jj}(k|k)}} \quad (5.4.2-7)$$

Under ideal conditions, the above is distributed  $\mathcal{N}(0, 1/N)$ .

Denoting by  $\xi$  a zero-mean unity-variance normal random variable, let  $r_1$  be such that

$$P\{\xi \in [-r_1, r_1]\} = 1 - \alpha \quad (5.4.2-8)$$

where, say,  $\alpha = 0.05$ . Then, since the standard deviation of  $\mu_j$  is  $1/\sqrt{N}$ , the  $1 - \alpha$  probability region for it will be  $[-r, r]$  where  $r = r_1/\sqrt{N}$ . For the above value of  $\alpha$  one has the two-sided 95% region given by  $r_1 = 1.96$ .

Thus, using this acceptance region based on the normal density, the hypothesis that the true mean of the state estimation error is zero is accepted if

$$\bar{\mu}_j(k) \in [-r, r] \quad (5.4.2-9)$$

The commensurateness of the innovations with their filter-calculated covariances — criterion (b) — is tested in a similar manner.

Under the hypothesis that the filter is consistent, the **normalized innovation squared (NIS)**

$$\epsilon_\nu(k) = \nu(k)' S(k)^{-1} \nu(k) \quad (5.4.2-10)$$

has a chi-square distribution with  $n_z$  degrees of freedom, where  $n_z$  is the dimension of the measurement.

From  $N$  independent samples  $\epsilon_\nu^i(k)$  one calculates the ( $N$ -run) **average NIS**

$$\bar{\epsilon}_\nu(k) = \frac{1}{N} \sum_{i=1}^N \epsilon_\nu^i(k) \quad (5.4.2-11)$$

which is then tested as in (5.4.2-5) but with acceptance region determined based on the fact that  $N\bar{\epsilon}_\nu(k)$  is chi-square distributed with  $Nn_z$  degrees of freedom.

Similarly to the procedure for state errors, if (5.4.2-11) is too large, then a bias test (i.e., whether the mean of the innovations is nonzero) has to be carried out.

The **whiteness test** for the innovations can be done for a single component at a time as follows.

The following ( $N$ -run) **sample autocorrelation** statistic is used

$$\bar{\rho}_l(k, j) = \sum_{i=1}^N \nu_l^i(k) \nu_l^i(j) \left[ \sum_{i=1}^N [\nu_l^i(k)]^2 \sum_{i=1}^N [\nu_l^i(j)]^2 \right]^{-1/2} \quad l = 1, \dots, n_z \quad (5.4.2-12)$$

For  $N$  large enough, a normal approximation of the density of (5.4.2-12) for  $k \neq j$  is convenient (and reasonable in view of the central limit theorem). If the innovations are zero mean and white, then the mean of (5.4.2-12) is zero and its variance is  $1/N$ . (See problem 5-4.)

Alternatively, it can be done simultaneously for the entire innovation vector by using the following statistic

$$\bar{\rho}(k, j) = \frac{1}{\sqrt{n_z}} \sum_{i=1}^N \nu^i(k)' \left[ \sum_{i=1}^N \nu^i(k) \nu^i(k)' \right]^{-1/2} \left[ \sum_{i=1}^N \nu^i(j) \nu^i(j)' \right]^{-1/2} \nu^i(j) \quad (5.4.2-13)$$

Similarly to (5.4.2-12), for  $N$  large enough, (5.4.2-13) for  $k \neq j$  is zero-mean with variance is  $1/N$  if the innovations are zero-mean and white (see problem 5-4) and a normal approximation can be used.

Thus, using the same acceptance region based on the normal density as in (5.4.2-9), the hypothesis that the true correlation of the innovation sequence is zero — criterion (c) — is accepted if

$$\bar{\rho}(k, j) \in [-r, r] \quad (5.4.2-14)$$

## Real-Time (Single-Run) Tests

All the above tests assume that  $N$  **independent runs** have been made. While they can be used on a single run ( $N = 1$ ), they have a very high variability in this case, as illustrated above. The question is whether one can achieve a low variability of the test statistic based on a single run, as in a real-time implementation — that is, having **real-time consistency tests**.

Test (a) requires that such independent simulations be made; however, criteria (b) and (c) can be tested on a **single run** in time as follows.

These tests are based on replacing the **ensemble averages** by **time averages** based on the *ergodicity* of the innovation sequence.

The whiteness test statistic for innovations, which are  $j$  steps apart, from a single run can be written as the **time-average autocorrelation**

$$\bar{\rho}_l(j) = \sum_{k=1}^K \nu_l(k)\nu_l(k+j) \left[ \sum_{k=1}^K \nu_l(k)^2 \sum_{k=1}^K \nu_l(k+j)^2 \right]^{-1/2} \quad l = 1, \dots, n_z \quad (5.4.2-15)$$

This statistic is, for large enough  $K$ , in view of the central limit theorem, normally distributed. Furthermore, similarly to (5.4.2-12), its variance can be shown to be  $1/K$  (see problem 5-4). Also, similarly to (5.4.2-13), one can write a single statistic that incorporates the entire innovation vector.

Criterion (b) can be tested with the **time-average normalized innovation squared** statistic

$$\bar{\epsilon}_\nu = \frac{1}{K} \sum_{k=1}^K \nu(k)'S(k)^{-1}\nu(k) \quad (5.4.2-16)$$

If the innovations are white, zero mean, and with covariance  $S(k)$ , then  $K\bar{\epsilon}_\nu$  has a chi-square distribution with  $Kn_z$  degrees of freedom.

The probability regions for (5.4.2-15) and (5.4.2-16) for acceptance of the “consistent filter” hypothesis are then set up as before.

### 5.4.3 Examples of Filter Consistency Testing

The example of Section 5.3 is continued to illustrate the use of the **consistency tests**. The following tests are carried out:

- Offline single-run (simulation) tests
- Offline multiple run (Monte Carlo simulation) tests
- Online single-run (real time) tests

Two cases will be considered:

1. A filter which is based on exactly the same model as the process — that is, **matched** to the system
2. A filter which is based on a different model than the system — that is, **mismatched** to the system

### Single-Run Simulation Tests

Figure 5.4.3-1 shows, for a *single run*, the behavior of the state's **NEES** (5.4.2-2) for various values of the process noise variance  $q$  for filters that are perfectly matched. Out of 100 points, 3 to 6 are found outside the 95% probability region.

In this case a one-sided region was considered. The upper limit of this probability region is approximately 6 since, for a  $n_x = 2$  degrees of freedom chi-square random variable, the 5% tail point is

$$\chi^2_2(0.95) = 5.99 \quad (5.4.3-1)$$

Note that in this case the two-sided 95% region is [0.05, 7.38]. Since the lower limit is practically zero, only the upper limit is of interest and it was taken for the 5% tail rather than for the 2.5% tail, which is 7.38. It should be noted that taking a 5% or a 2.5% (or a 1%) tail is rather arbitrary.

### Monte Carlo Simulation Tests

Figures 5.4.3-2a to 5.4.3-2c illustrate, using DynaEst<sup>TM</sup>, the test statistics obtained from  $N = 50$  Monte Carlo runs. Two-sided probability regions are used in the sequel.

Figure 5.4.3-2a shows the state's *N-run average NEES* (5.4.2-4). The two-sided 95% region for a 100 degrees of freedom chi-square random variable is

$$[\chi^2_{100}(0.025), \chi^2_{100}(0.975)] = [74.2, 129.6] \quad (5.4.3-2)$$

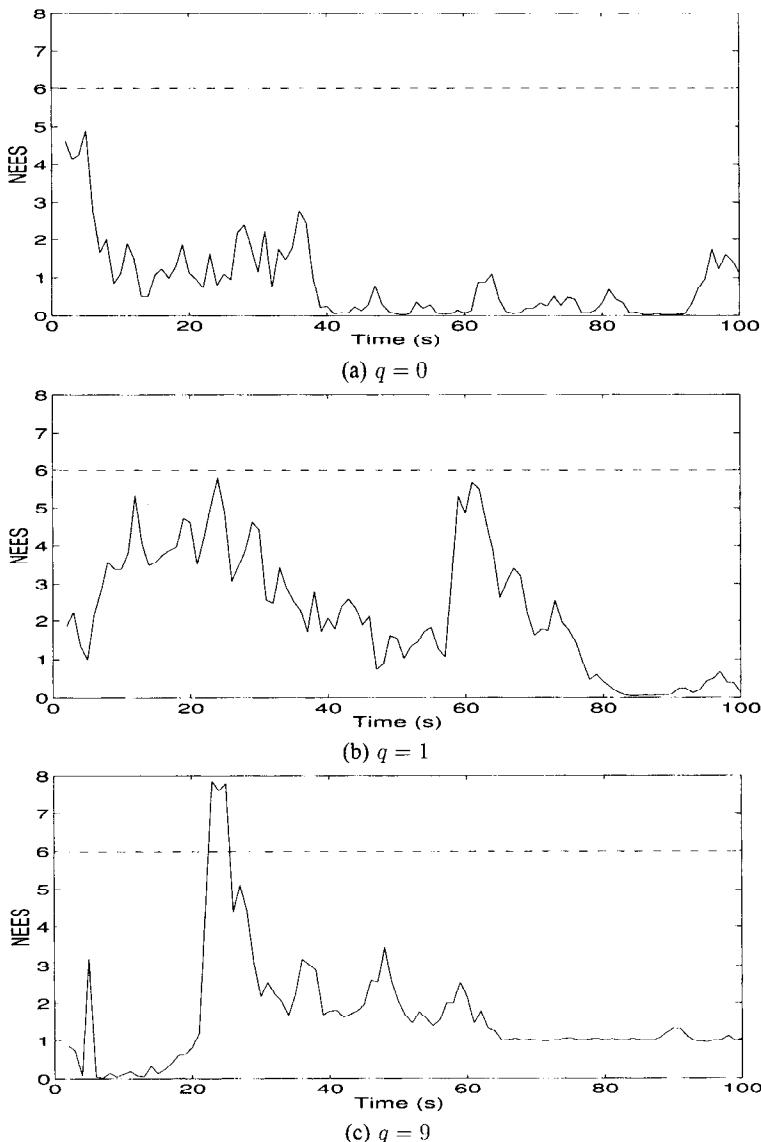
Dividing the above by  $N = 50$ , the 95% probability region (5.4.2-5) for the average normalized state estimation error squared becomes [1.5, 2.6]. Note that 6 out of the 100 points fall outside this 95% region, which is acceptable.

Figure 5.4.3-2b shows the (*N-run*) *average NIS* (5.4.2-11). Noting that the innovations are scalar, the 95% probability region will be based on the 50 degrees of freedom chi-square distribution, and is

$$[\chi^2_{50}(0.025), \chi^2_{50}(0.975)] = [32.3, 71.4] \quad (5.4.3-3)$$

Dividing by  $N = 50$ , the region becomes [0.65, 1.43]. As the plot shows, 4 out of the 100 points are outside the 95% region, again an acceptable situation.

Figure 5.4.3-2c shows the (*N-run*) *sample autocorrelation* of the innovations (5.4.2-12) one step apart ( $k - j = 1$ ). The 95% region  $[-1.96\sigma, 1.96\sigma]$  is, for  $\sigma = \frac{1}{\sqrt{N}} = 0.141$ , the interval  $[-0.277, 0.277]$ . In this case, 8 out of the 100 points fall outside the 95% region, which is also acceptable.



**Figure 5.4.3-1:** Normalized state estimation error squared from a single run with its 95% probability region.

## Remark

These probability regions are also called **acceptance regions** because, if the test statistics fall in these regions, one can *accept* the hypothesis that the *filter is consistent*.

## A Mismatched Filter

Next, a **mismatched filter** is examined. For this purpose it is assumed that the true (i.e., system) process noise has variance  $q = 9$  while the model (i.e., filter) process noise has variance  $q_F = 1$ .

The normalized state estimation error squared, obtained with DynaEst<sup>TM</sup>, is shown in Fig. 5.4.3-3. The mismatch caused 41 points out of 100 to be outside the 95% probability region in a single run, clearly an unacceptable situation. The results of the Monte Carlo runs ( $N = 50$ ) shown in Figs. 5.4.3-3b and 5.4.3-4 all show the serious mismatch: All points are outside the 95% region — actually they are all in the upper 2.5% tail region.

## Real-Time Tests

Finally, the **real-time consistency tests** — that is, the *single-run tests that can be performed in real time* — are presented. For the correct filter the **time-average autocorrelation** of the innovations (5.4.2-15) obtained from 100 samples in time was

$$\bar{\rho}(1) = 0.152 \quad (5.4.3-4)$$

Under the assumption that the filter is correct, the error in the above estimate is normally distributed with mean zero and variance  $1/100$

$$\bar{\rho} - \rho \sim \mathcal{N}(0, 0.1^2) \quad (5.4.3-5)$$

and its 95% probability region is  $[-0.196, 0.196]$ . The estimate (5.4.3-4) falls in this region, as expected, since the filter is matched to the system.

In the mismatched case the estimate was obtained as

$$\bar{\rho}(1) = 0.509 \quad (5.4.3-6)$$

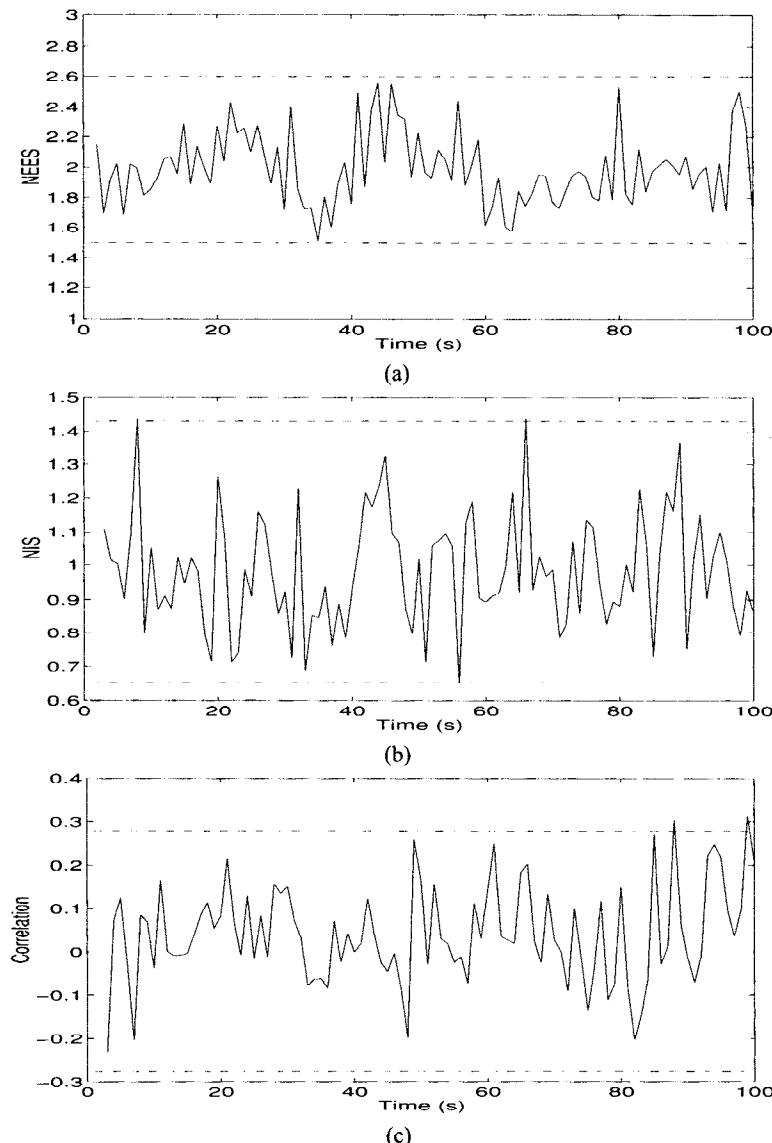
which is clearly much too large and outside the region.

The other single-run test is for the **time-average normalized innovation squared** (5.4.2-16), also over 100 time steps.

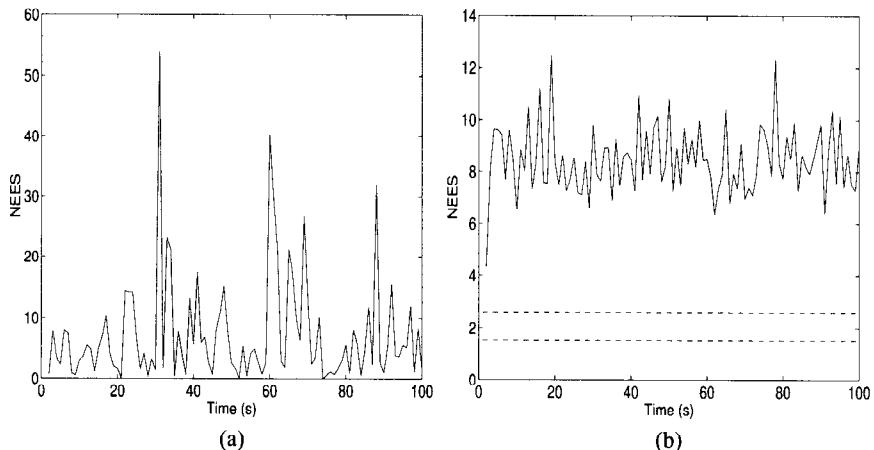
For the matched filter, the result was

$$\bar{\epsilon}_\nu = 0.936 \quad (5.4.3-7)$$

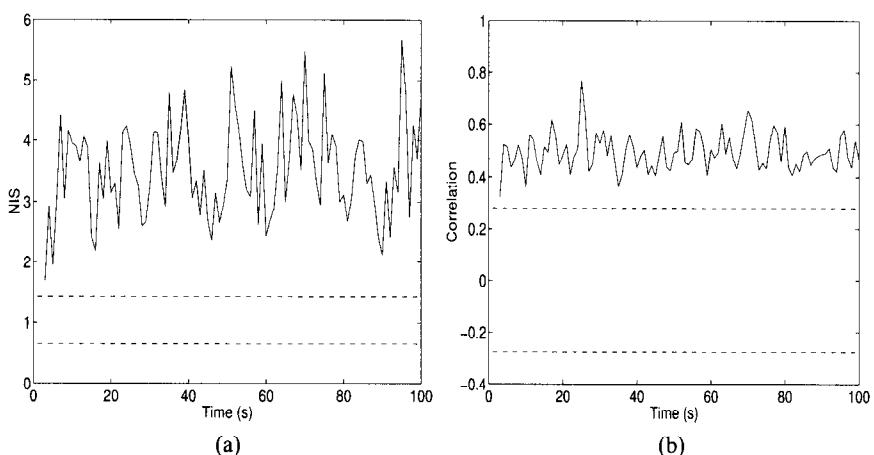
(the ideal value is 1). The 95% confidence region is, based on the 100 degrees of freedom chi-square distribution, the interval  $[0.74, 1.3]$ .



**Figure 5.4.3-2:** (a) Normalized state estimation error squared, (b) normalized innovation squared, (c) innovation autocorrelation, from 50 Monte Carlo runs with their 95% probability regions for  $q = 1$ .



**Figure 5.4.3-3:** Mismatched filter: (a) normalized state estimation error squared from a single run, (b) 50-run Monte Carlo average with its 95% probability region.



**Figure 5.4.3-4:** Mismatched filter: (a) normalized innovation squared, (b) innovation autocorrelation, from 50 Monte Carlo runs with their 95% probability regions.

For the mismatched filter, the result was

$$\bar{\epsilon}_v = 2.66 \quad (5.4.3-8)$$

which is clearly unacceptable.

The above illustrates how one can detect filter mismatch in a single run using suitable statistics and their probability regions.

## Filter Tuning

The process noise is used in practice to model disturbances — for example, unknown inputs like target maneuvers in tracking. The procedure to match the process noise variance to suitably model such disturbances is called **filter tuning**. Since such inputs are not real noise in the probabilistic sense, it is said that the filter uses **pseudo-noise** or **artificial noise**.

The procedure for tuning is to make the filter consistent, that is, the three criteria (a)–(c) from Subsection 5.4.2 should be satisfied. While this is easier said than done, in practice one has to strive to make the filter as close to being consistent as feasible while at the same time achieving small RMS estimation errors. Note that the RMS errors are **unnormalized errors**, while the test statistics are normalized — that is, divided (in a matrix sense) by the filter-calculated variances.

This is discussed in more detail in Subsection 11.6.7 and Section 11.7, where the design of several filters for a realistic situation is illustrated.

### 5.4.4 Absolute Errors

In all problems one is also (actually even more) interested in the absolute performance of a filter, which is evaluated by the **RMS errors**. In the following, the expressions for these errors in **position**, **velocity**, **speed**,<sup>3</sup> and **course**<sup>4</sup> are given.

For a scalar  $\xi$ , the **RMS error from N Monte Carlo runs** with observed error  $\tilde{\xi}_i$  in run  $i$  is

$$\text{RMS}(\tilde{\xi}) = \sqrt{\frac{1}{N} \sum_{i=1}^N \tilde{\xi}_i^2} \quad (5.4.4-1)$$

For a position vector  $x = [\xi \ \eta]$ , the RMS position error is

$$\text{RMS}(\tilde{x}) = \sqrt{\frac{1}{N} \sum_{i=1}^N (\tilde{\xi}_i^2 + \tilde{\eta}_i^2)} \quad (5.4.4-2)$$

<sup>3</sup>Speed, often confused with velocity, is the **magnitude** of the velocity, which is a vector.

<sup>4</sup>Course is the direction of the velocity vector, typically in the horizontal plane. The term **heading** is often used incorrectly in its place. Heading is the pointing direction of the aircraft or vessel — the direction you think you are going — while course is the direction you are actually going. This is relevant in motion in fluid media, like air or water, where winds aloft or currents can be significant.

Similarly, for the velocity  $\dot{x} = [\dot{\xi} \ \dot{\eta}]$ , the RMS velocity error is

$$\text{RMS}(\tilde{x}) = \sqrt{\frac{1}{N} \sum_{i=1}^N (\tilde{\xi}_i^2 + \tilde{\eta}_i^2)} \quad (5.4.4-3)$$

For the speed

$$u_i = \sqrt{\dot{\xi}_i^2 + \dot{\eta}_i^2} \quad (5.4.4-4)$$

estimated as

$$\hat{u}_i = \sqrt{\dot{\xi}_i^2 + \dot{\eta}_i^2} \quad (5.4.4-5)$$

the RMS error is given, according to (5.4.4-1), by, with  $\tilde{u}_i = u_i - \hat{u}_i$ ,

$$\text{RMS}(\tilde{u}) = \sqrt{\frac{1}{N} \sum_{i=1}^N \tilde{u}_i^2} \quad (5.4.4-6)$$

Similarly, for the course (measured from the  $\eta$  axis (North) clockwise)

$$\theta_i = \tan^{-1} \left[ \frac{\dot{\xi}_i}{\dot{\eta}_i} \right] \quad (5.4.4-7)$$

estimated as

$$\hat{\theta}_i = \tan^{-1} \left[ \frac{\dot{\xi}_i}{\dot{\eta}_i} \right] \quad (5.4.4-8)$$

the RMS error is given, according to (5.4.4-1), by, with  $\tilde{\theta}_i = \theta_i - \hat{\theta}_i$ ,

$$\text{RMS}(\tilde{\theta}) = \sqrt{\frac{1}{N} \sum_{i=1}^N \tilde{\theta}_i^2} \quad (5.4.4-9)$$

## 5.4.5 Filter Consistency — Summary

*Consistency testing* is crucial for *estimator optimality evaluation*.

A state estimator is *consistent* if the first- and second-order moments of its estimation errors are as the theory predicts:

- Their means are zero — the estimates are unbiased.
- Their covariance matrices are as calculated by the filter.

The statistic that tests the mean and the covariance is the *normalized estimation error squared (NEES)*. This is done for

- the state and
- the innovations (measurement prediction errors).

Under the Gaussian assumption, these statistics are chi-square distributed and should be with a high probability in certain intervals — the corresponding probability regions. These tests are also called “chi-square” tests.

Bias in the estimates, if significant, or error magnitudes too large compared to the filter-calculated standard deviations will be detected by this statistic.

This test can be used as follows:

- Offline (in simulations) for state estimation errors — the truth is available for comparison.
- Offline or online (in real time) for the innovation.

The test for the *innovation’s whiteness* is the *sample autocorrelation* whose magnitude has to be below a certain threshold. This can be carried out offline or online.

These tests become very powerful; that is, they can detect *inconsistent (mismatched)* filters when used in Monte Carlo runs.

Monte Carlo runs evaluate *ensemble averages* of the test statistics. The averages obtained from *independent runs* — with *all the random variables independent from run to run* — decrease the variability and therefore increase the power of the tests.

Acceptance regions — *upper and lower bounds* — within which the test statistics should be for consistent filters, have been established.

If the normalized error statistic exceeds the upper bound, then

- there is *significant bias* in the estimates, or
- the *errors are too large* compared to the filter-calculated covariance, or
- the *covariance is too small*.

In this case the filter is *optimistic*.

If the normalized error statistic is below the lower bound, then the *covariance is too large*. In this case the filter can be said to be *pessimistic*.

For *real-time filter performance monitoring*, one can implement tests based on *time averaging* for innovation magnitude and whiteness.

## 5.5 INITIALIZATION OF STATE ESTIMATORS

### 5.5.1 Initialization and Consistency

A state estimation filter is called consistent if its estimation errors are “com-mensurate” or “compatible” with the filter-calculated covariances. At *initialization*, it is just as important that the covariance associated with the initial estimate reflects realistically its accuracy.

According to the Bayesian model, the *true initial state is a random variable*, assumed to be normally distributed with a known mean — the initial estimate

— and a given covariance matrix

$$x(0) \sim \mathcal{N}[\hat{x}(0|0), P(0|0)] \quad (5.5.1-1)$$

The norm (“chi-square”) test for the initial estimation error is

$$\tilde{x}(0|0)'P(0|0)^{-1}\tilde{x}(0|0) \leq c_1 \quad (5.5.1-2)$$

where  $c_1$  is the upper limit of the, say, 95% confidence region from the chi-square distribution with the corresponding number of degrees of freedom.

Sometimes one has to choose the initial covariance — the “choice” has to be such that (5.5.1-2) is satisfied. A large error in the initial estimate, if the latter is deemed highly accurate, will persist a long time because it leads to a low filter gain and thus the new information from the measurements receives weighting that is too low. (See problem 5-6.)

For the one-dimensional case, (5.5.1-2) can be stated as follows: The initial error should be not more than, say, two times the associated standard deviation — this is called a “ $2\sigma$ ” error. If the initial variance is such that the initial error is “ $1\sigma$ ” and the subsequent measurements are accurate, then the initial error will decrease rapidly.

## 5.5.2 Initialization in Simulations

As discussed before, the Bayesian model for initialization is as follows:

- The initial state is a random variable.
- The (prior) pdf of the initial state is known and assumed Gaussian.

Thus, in simulations one should generate the initial state with a random number generator according to (5.5.1-1). The initial estimate is, again according to the Bayesian model, a *known quantity* together with the initial covariance matrix.

This approach, while rigorous according to the assumptions of the filter, is not so appealing because the true initial state, which defines the *scenario*, will be different in each simulation.

A more appealing approach is the following:

- Choose the initial true state.
- Generate the initial estimate according to

$$\hat{x}(0|0) \sim \mathcal{N}[x(0), P(0|0)] \quad (5.5.2-1)$$

The above relies on the fact that, switching the roles of the initial state  $x(0)$  and the initial estimate  $\hat{x}(0|0)$ , one has the algebraic identity

$$\begin{aligned} p[x(0)|\hat{x}(0|0)] &= \mathcal{N}[x(0); \hat{x}(0|0), P(0|0)] \\ &= \mathcal{N}[\hat{x}(0|0); x(0), P(0|0)] \\ &= p[\hat{x}(0|0)|x(0)] \end{aligned} \quad (5.5.2-2)$$

in view of the special form of the normal distribution. In other words, the *scenario*  $x(0)$  is *fixed* and the *initial condition*  $\hat{x}(0|0)$  of the filter is *random*.

### 5.5.3 A Practical Implementation in Tracking

The practical implementation of the initialization (5.5.2-1) can be done as follows. Consider two state components, say, position  $\xi$  and velocity  $\dot{\xi}$  in a given coordinate. If only position measurements

$$z(k) = \xi(k) + w(k) \quad (5.5.3-1)$$

are available, then for the true values  $\xi(k)$ ,  $k = -1, 0$ , one generates the corresponding measurement noises, say

$$w(k) \sim \mathcal{N}[0, R] \quad (5.5.3-2)$$

Then, denoting by  $T$  the sampling interval, one has

$$\hat{\xi}(0|0) = z(0) \quad (5.5.3-3)$$

$$\hat{\dot{\xi}}(0|0) = \frac{z(0) - z(-1)}{T} \quad (5.5.3-4)$$

and the corresponding  $2 \times 2$  block of the initial covariance matrix is then<sup>5</sup>

$$P(0|0) = \begin{bmatrix} R & R/T \\ R/T & 2R/T^2 \end{bmatrix} \quad (5.5.3-5)$$

This method, called **two-point differencing**, guarantees consistency of the initialization of the filter, which starts updating the state at  $k = 1$ .

If several (Monte Carlo) runs are made, then the same initialization procedure has to be followed with new (*independent*) noises generated in every run according to (5.5.3-2). “Reuse” of the same initial conditions in Monte Carlo runs will lead to biased estimates (see problem 5-6).

#### Remark

The above amounts to a **first-order polynomial fitting** of the first two measurements. If (and only if) there are *significant* higher derivatives, then one should use more than two points and a **higher-order polynomial fitting** via LS is to be carried out.

#### One-Point Initialization

In some problems the variance of the velocity obtained from two-point differencing might be much larger than some known bound on the speed. In such a case one can initialize the position from a single observation and use zero as initial estimate for the velocity with an associated standard deviation equal to, say, half of the known maximum speed.

Note, however, that this will cause a bias in Monte Carlo runs for a fixed scenario, since the initial velocity error is fixed rather than random and zero-mean. Appropriate care should be exercised when interpreting the results.

---

<sup>5</sup>Neglecting the process noise (see problem 5-13).

## Automatic Start-up Without Initial Estimate

A general and exact technique for start-up of a filter without initial estimates via the *information filter* form of the KF is discussed in Subsection 7.2.3.

### 5.5.4 Filter Initialization — Summary

The error in the initial state estimate has to be *consistent* with the initial state covariance. When an initial covariance is “chosen,” it should be such that the error is at most 2 times the corresponding standard deviation.

According to the Bayesian assumptions in the Kalman filter,

- *the true initial state is a random variable and*
- *the initial estimate is a fixed known quantity.*

In simulations one can reverse this point of view by fixing the initial state (the scenario) and generating the initial estimate with a random number generator as follows:

The initial estimate is generated with mean equal to the true initial state and with covariance equal to the initial state covariance.

In practice, the initialization can be done from two consecutive position measurements by

1. using the latest measurement as initial position estimate,
2. differencing them to obtain the velocity estimate,
3. calculating the corresponding covariance matrix.

This amounts to a first-order polynomial fit. Higher-order polynomial fits can also be used for initializing the estimation of states that have higher derivatives.

This initialization method should also be followed in Monte Carlo simulations where, with *noises independent from run to run*, one will then obtain initial errors also independent from run to run.

In some circumstances, one-point initialization is appropriate.

There is a general procedure to initialize filters, to be discussed later.

## 5.6 SENSITIVITY

The derivation of the optimal filter gain in (5.2.3-11) assumes that the mathematical description of the system given by (5.2.1-1)–(5.2.1-4) is exact. That is, the system matrices and the noise statistics used in the Kalman filter model match those of the system. It also assumes that the chosen filter state fully describes the system state.

In practice, exact information about the system is not available, and therefore the model assumed by the filter designer is different from the truth. In some cases, in order to reduce computational load, the designer might implement a reduced-order filter with fewer states than in the “full model” or use suboptimal gains which are more easily computed than the optimal one.

In view of the modeling errors and implementation approximations, **sensitivity analysis**, where the sensitivity of the filter to any such mismatch is determined, is in order.

### 5.6.1 Model Mismatch

Let the model assumed by the filter be described by

$$x_f(k+1) = F_f(k)x_f(k) + G_f(k)u_f(k) + v_f(k) \quad (5.6.1-1)$$

and let the measurement process be described by

$$z_f(k) = H_f(k)x_f(k) + w_f(k) \quad (5.6.1-2)$$

where  $x_f(k)$  is the filter state and the mutually independent model noise components  $v_f(k)$  and  $w_f(k)$  are

$$v_f(k) \sim \mathcal{N}(0, Q_f(k)) \quad (5.6.1-3)$$

and

$$w_f(k) \sim \mathcal{N}(0, R_f(k)) \quad (5.6.1-4)$$

Clearly, these are different from the actual system equations defined by the system matrices  $F(k)$ ,  $G(k)$  and  $H(k)$  and by the error statistics  $Q(k)$  and  $R(k)$ . As a result, the “optimal” gain  $W(k+1)$  evaluated by the filter, which in this case is given by

$$\begin{aligned} W(k+1) &= P_f(k+1|k)H_f(k+1)' \\ &\cdot [H_f(k+1)P_f(k+1|k)H_f(k+1)' + R_f(k+1)]^{-1} \end{aligned} \quad (5.6.1-5)$$

which yields the associated **filter-calculated covariance matrices**

$$P_f(k+1|k) = F_f(k)P_f(k|k)F_f(k)' + Q_f(k) \quad (5.6.1-6)$$

and, using the Joseph form of the covariance update (5.2.3-18), which is valid for arbitrary filter gain,

$$\begin{aligned} P_f(k+1|k+1) &= [I - W(k+1)H_f(k+1)]P_f(k+1|k) \\ &\cdot [I - W(k+1)H_f(k+1)]' + W(k+1)R_f(k+1)W(k+1)' \end{aligned} \quad (5.6.1-7)$$

is not really optimal for the system given by (5.2.1-1)–(5.2.1-4), but *only* for the filter model assumed in (5.6.1-1)–(5.6.1-4). With these values, the state estimate  $\hat{x}(k+1|k+1)$  is given by

$$\hat{x}(k+1|k) = F_f(k)\hat{x}(k|k) + G_f(k)u_f(k) \quad (5.6.1-8)$$

$$\begin{aligned} \hat{x}(k+1|k+1) &= \hat{x}(k+1|k) + W(k+1) \\ &\cdot [z(k+1) - H_f(k+1)\hat{x}(k+1|k)] \end{aligned} \quad (5.6.1-9)$$

To analyze the evolution of the error between the actual state  $x(k)$  and the filter estimated state  $\hat{x}(k|k)$ , define the following:

$$\tilde{x}(k+1|k) = x(k+1) - \hat{x}(k+1|k) \quad (5.6.1-10)$$

$$\tilde{x}(k+1|k+1) = x(k+1) - \hat{x}(k+1|k+1) \quad (5.6.1-11)$$

which are the actual errors in the predicted and updated state, respectively, with associated **MSE matrices**<sup>6</sup>  $P(k+1|k)$  and  $P(k+1|k+1)$ . Also define the following model mismatches:

$$\tilde{F}(k) = F(k) - F_f(k) \quad (5.6.1-12)$$

$$\tilde{H}(k) = H(k) - H_f(k) \quad (5.6.1-13)$$

Then the state prediction error can be written as<sup>7</sup>

$$\begin{aligned} \tilde{x}(k+1|k) &= x(k+1) - \hat{x}(k+1|k) = F(k)x(k) + v(k) - F_f\hat{x}(k|k) \\ &= F_f(k)\tilde{x}(k|k) + \tilde{F}(k)x(k) + v(k) \end{aligned} \quad (5.6.1-14)$$

Combining this with (5.2.1-1), one can define an augmented state equation

$$\begin{bmatrix} x(k+1) \\ \tilde{x}(k+1|k) \end{bmatrix} = \begin{bmatrix} F(k) & 0 \\ \tilde{F}(k) & F_f(k) \end{bmatrix} \begin{bmatrix} x(k) \\ \tilde{x}(k|k) \end{bmatrix} + \begin{bmatrix} I_{n_x \times n_x} \\ I_{n_x \times n_x} \end{bmatrix} v(k) \quad (5.6.1-15)$$

where  $I_{n_x \times n_x}$  is an  $n_x \times n_x$  identity matrix.

Since one is interested in the actual error statistics in the presence of modeling errors, namely,  $P(k+1|k)$  and  $P(k+1|k+1)$ , define the following **mean-square value (MSV) matrices**, also called **correlation matrices**<sup>8</sup>:

$$E \left\{ \begin{bmatrix} x(k) \\ \tilde{x}(k|k) \end{bmatrix} \begin{bmatrix} x(k) \\ \tilde{x}(k|k) \end{bmatrix}' \right\} \triangleq \begin{bmatrix} X(k) & C(k|k)' \\ C(k|k) & P(k|k) \end{bmatrix} \quad (5.6.1-16)$$

and

$$E \left\{ \begin{bmatrix} x(k+1) \\ \tilde{x}(k+1|k) \end{bmatrix} \begin{bmatrix} x(k+1) \\ \tilde{x}(k+1|k) \end{bmatrix}' \right\} \triangleq \begin{bmatrix} X(k+1) & C(k+1|k)' \\ C(k+1|k) & P(k+1|k) \end{bmatrix} \quad (5.6.1-17)$$

In the above,  $X$  is the MSV matrix of the state vector,  $P$  is an estimation MSE, and  $C$  is a cross-term.

<sup>6</sup>In can be shown that, usually, the actual errors  $\tilde{x}(k+1|k)$  and  $\tilde{x}(k+1|k+1)$  are not zero-mean in the presence of modeling errors, resulting in biased estimates. Thus, the term MSE matrices of these quantities is used in place of covariance matrices.

<sup>7</sup>Without loss of generality, in the sequel it is assumed that  $u(k) = u_f(k) = 0$ . The derivation can be easily generalized for the more general case, resulting in longer equations.

<sup>8</sup>Using the terminology from stochastic processes, this is the autocorrelation (or correlation) of the corresponding random sequence at a single point in time, or its MS value, thus the term MSV matrix.

Then

$$\begin{bmatrix} X(k+1) & C(k+1|k)' \\ C(k+1|k) & P(k+1|k) \end{bmatrix} = \begin{bmatrix} F(k) & 0 \\ \tilde{F}(k) & F_f(k) \end{bmatrix} \begin{bmatrix} X(k) & C(k|k)' \\ C(k|k) & P(k|k) \end{bmatrix} \\ \cdot \begin{bmatrix} F(k) & 0 \\ \tilde{F}(k) & F_f(k) \end{bmatrix}' + \begin{bmatrix} Q(k) & Q(k) \\ Q(k) & Q(k) \end{bmatrix} \quad (5.6.1-18)$$

which yields

$$\begin{aligned} P(k+1|k) &= F_f(k)P(k|k)F_f(k)' + Q(k) + \tilde{F}(k)X(k)\tilde{F}(k)' \\ &\quad + F_f(k)C(k|k)\tilde{F}(k)' + \tilde{F}(k)C(k|k)'F_f(k)' \end{aligned} \quad (5.6.1-19)$$

Similarly, the state update error defined in (5.6.1-11) is given by

$$\begin{aligned} \tilde{x}(k+1|k+1) &= x(k+1) - \hat{x}(k+1|k) - W(k+1)[z(k+1) \\ &\quad - H_f(k+1)\hat{x}(k+1|k)] \end{aligned} \quad (5.6.1-20)$$

which can be rewritten as

$$\begin{bmatrix} x(k+1) \\ \tilde{x}(k+1|k+1) \end{bmatrix} = K(k+1) \begin{bmatrix} x(k+1) \\ \tilde{x}(k+1|k) \end{bmatrix} \\ + \begin{bmatrix} 0 \\ -W(k+1) \end{bmatrix} w(k+1) \quad (5.6.1-21)$$

where

$$K(k+1) \triangleq \begin{bmatrix} I & 0 \\ -W(k+1)\tilde{H}(k+1) & I - W(k+1)H_f(k+1) \end{bmatrix} \quad (5.6.1-22)$$

with associated MSV matrix

$$\begin{bmatrix} X(k+1) & C(k+1|k+1)' \\ C(k+1|k+1) & P(k+1|k+1) \end{bmatrix} = K(k+1) \\ \cdot \begin{bmatrix} X(k+1) & C(k+1|k)' \\ C(k+1|k) & P(k+1|k) \end{bmatrix} K(k+1)' \\ + \begin{bmatrix} 0 & 0 \\ 0 & W(k+1)R(k+1)W(k+1)' \end{bmatrix} \quad (5.6.1-23)$$

Then the actual updated MSE recursion equation is given by

$$\begin{aligned} P(k+1|k+1) &= [I - W(k+1)H_f(k+1)]P(k+1|k) \\ &\quad [I - W(k+1)H_f(k+1)]' + W(k+1)R(k+1)W(k+1)' \\ &\quad + W(k+1)\tilde{H}(k+1)X(k+1)\tilde{H}(k+1)'W(k+1)' \\ &\quad - [I - W(k+1)H_f(k+1)]C(k+1|k)\tilde{H}(k+1)'W(k+1)' \\ &\quad - W(k+1)\tilde{H}(k+1)C(k+1|k)'[I - W(k+1)H_f(k+1)] \end{aligned} \quad (5.6.1-24)$$

together with

$$\begin{aligned} C(k+1|k+1) &= [I - W(k+1)H_f(k+1)]C(k+1|k) \\ &\quad - W(k+1)\tilde{H}(k+1)X(k+1) \end{aligned} \quad (5.6.1-25)$$

The actual MSE matrices  $P(k+1|k)$  and  $P(k+1|k+1)$  given by (5.6.1-19) and (5.6.1-24) can be compared with the filter calculated covariance matrices  $P_f(k+1|k)$  and  $P_f(k+1|k+1)$  in (5.6.1-6) and (5.6.1-7), respectively. It can be seen that the actual MSE matrices are “larger” than the filter calculated covariance matrices — the increased uncertainty accounts for the modeling errors. Also, in the absence of modeling errors, the filter calculated covariances are the same as the actual MSE.

Note that the calculation of actual MSE matrices requires the knowledge of true system matrices  $F(k)$ ,  $H(k)$ ,  $Q(k)$  and  $R(k)$ , which are not always available. When true system matrices are not available, their nominal values, which represent an “average” scenario for the problem under consideration, can be used.

The auxiliary MSV matrices  $X(k)$ ,  $C(k+1|k)$  and  $C(k|k)$  are evaluated only to update the actual MSE matrices  $P(k+1|k)$  and  $P(k|k)$ . Their initialization is carried out as follows:

$$X(0) = E[x(0)x(0)'] = \hat{x}(0|0)\hat{x}(0|0)' + P(0|0) \quad (5.6.1-26)$$

and

$$C(0|0) = P(0|0) = P_0 \quad (5.6.1-27)$$

where  $P_0$  is the actual MSE of the initial state estimate. For the filter covariance calculation the initial state covariance is  $P_f(0|0)$ , which can be different from  $P(0|0)$ . This is relevant, e.g., for the evaluation of the effect of an incorrect initial covariance (see problem 5-6).

Figure 5.6.1-1 shows the flowchart of a cycle in a model-mismatched Kalman filter together with the evaluation of the actual MSE matrices.

### Remark

Another point of interest is the asymptotic behavior of the actual errors. For example, by combining (5.6.1-14) and (5.6.1-21), the evolution of the updated state estimation error can be written as

$$\begin{aligned} \tilde{x}(k+1|k+1) &= [I - W(k+1)H_f(k+1)]F_f(k)\tilde{x}(k|k) \\ &\quad - W(k+1)\tilde{H}(k+1)x(k+1) \\ &\quad + [I - W(k+1)H_f(k+1)]\left[\tilde{F}(k)x(k) + v(k)\right] \end{aligned} \quad (5.6.1-28)$$

It can be noted that the above error system depends on the true state  $x(k)$  in a multiplicative manner, which indicates that the error can become unbounded for unbounded true state. Thus, modeling errors can result in unbounded estimation errors.

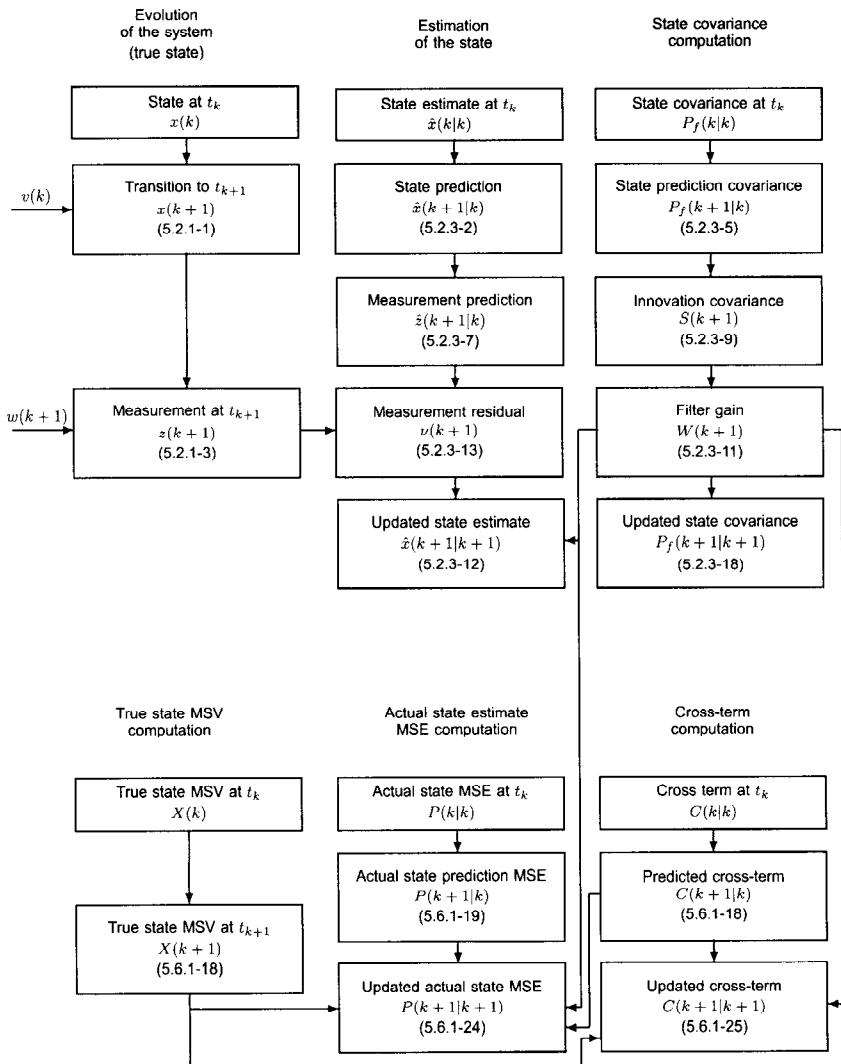


Figure 5.6.1-1: One cycle in a model-mismatched Kalman filter.

## 5.6.2 Reduced-Order Filters

A special case of modeling inaccuracy is when some components of the actual state vector are ignored and a Kalman filter to estimate the retained state components is implemented. This reduced-order filter may be due to incomplete knowledge about the additional states or because of the need for reduced computational load. For example, the “nearly constant velocity” model (see Section 6.4), which assumes that the target under track evolves at almost a constant velocity, is commonly used in tracking applications. This effectively ignores the (unknown) higher-order components, for example, the acceleration and jerk.

Let the size of the true state vector  $x(k)$  in (5.2.1-1) be  $n$  and that of the reduced order state used in the filter be  $n_r$ . Without loss of generality, assume that filter state  $x_r(k)$  consists of the first  $n_r$  elements of  $x(k)$ . That is,

$$x_r(k) = [ I_{n_r \times n_r} \quad \mathbf{0}_{n_r \times n-n_r} ] x(k) = \mathcal{T}x(k) \quad (5.6.2-1)$$

Then

$$\begin{aligned} x_r(k+1) &= \mathcal{T}x(k+1) = \mathcal{T}F(k)\mathcal{T}'x_r(k) + \mathcal{T}v(k) \\ &= F_r(k)x_r(k) + v_r(k) \end{aligned} \quad (5.6.2-2)$$

and, since, typically, the neglected state components do not appear in the measurements,

$$z(k) = H_r(k)x_r(k) + w(k) = H_r(k)\mathcal{T}x(k) + w(k) \quad (5.6.2-3)$$

Note that the full-dimensional state estimate (with the last  $n - n_r$  components zero) is

$$\hat{x}_f = \mathcal{T}'\hat{x}_r \quad (5.6.2-4)$$

Its state transition matrix is, using (5.6.2-1) and (5.6.2-3) in (5.6.2-2), given by

$$F_f(k) = \mathcal{T}'F_r(k)\mathcal{T} \quad (5.6.2-5)$$

and its measurement matrix is, from (5.6.2-3),

$$H_f(k) = H_r(k)\mathcal{T} \quad (5.6.2-6)$$

The prediction and the update filter errors, which were defined in (5.6.1-10) and (5.6.1-11) for general modeling errors, respectively, are given by

$$\begin{aligned} \tilde{x}(k+1|k) &= x(k+1) - \hat{x}_f(k+1|k) \\ &= x(k+1) - \mathcal{T}'\hat{x}_r(k+1|k) \end{aligned} \quad (5.6.2-7)$$

$$\begin{aligned} \tilde{x}(k+1|k+1) &= x(k+1) - \hat{x}_f(k+1|k+1) \\ &= x(k+1) - \mathcal{T}'\hat{x}_r(k+1|k+1) \end{aligned} \quad (5.6.2-8)$$

Also, define the model mismatches, similar to (5.6.1-12) and (5.6.1-13), as

$$\tilde{F}(k) = F(k) - F_f(k) = F(k) - T' F_r(k) T \quad (5.6.2-9)$$

$$\tilde{H}(k) = H(k) - H_f(k) = H(k) - H_r(k) T \quad (5.6.2-10)$$

With these values, the actual prediction and update MSE matrices can be obtained by substituting (5.6.2-5) and (5.6.2-6) into (5.6.1-18) – (5.6.1-25).

In deciding whether to use a reduced-order filter, one needs to be satisfied that the difference between the resulting actual error statistics and the full-state filter's error statistics is indeed acceptable.

## Decoupling

Reduced-order filtering can be used to decouple the state estimates and implement the estimator using two or more reduced-order filters. This is especially useful when  $F(k)$  has a block-diagonal structure, or is approximately block-diagonal. For example, assume that the state vector  $x(k)$  (of size  $n_1 + n_2$ ) can be written in terms of two decoupled state subvectors  $x_1(k)$  and  $x_2(k)$  (of sizes  $n_1$  and  $n_2$ , respectively) as

$$\begin{aligned} x(k+1) &= \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} \\ &= \begin{bmatrix} F_1(k) & 0 \\ 0 & F_2(k) \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} v_1(k) \\ v_2(k) \end{bmatrix} \end{aligned} \quad (5.6.2-11)$$

and

$$\begin{aligned} z(k+1) &= \begin{bmatrix} z_1(k+1) \\ z_2(k+1) \end{bmatrix} \\ &= \begin{bmatrix} H_1(k) & 0 \\ 0 & H_2(k) \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix} \end{aligned} \quad (5.6.2-12)$$

The estimator for the above decoupled system can be implemented with two filters, one estimating  $x_1(k)$  and the other  $x_2(k)$ . The motivation for decoupling comes from the “divide-and-conquer” philosophy — the decoupled system’s computations complexity is of the order  $n_1^3 + n_2^3$  as opposed to  $(n_1 + n_2)^3$ .

An application of decoupled filtering is found in air traffic surveillance, where, usually, an aircraft’s motion in the horizontal place is independent of its vertical motion. In that case, the aircraft motion can be tracked using one filter for estimating the horizontal state (for example, positions and velocities in the North and East directions) and another for the vertical state (the altitude and the vertical velocity; see, e.g., [Yeddanapudi97, Wang99]).

### 5.6.3 Suboptimal Gains

Another common approximation is to use suboptimal filter gains, often precomputed or fixed ones, in order to reduce the computation load in evaluating the optimal gain  $W(k)$  for each update. While the optimal gain can be precomputed offline as shown in Fig. 5.2.4-1, storing the entire gain history may not be practicable due to high storage requirements.

Two possible approximations are either to use a fixed gain, usually the steady-state gain  $W_\infty$ , or to use a piecewise linear approximation of the optimal gain. In the former case, the covariance update in (5.6.1-7), which is valid for any arbitrary gain, becomes

$$\begin{aligned} P_f(k+1|k+1) &= [I - W_\infty H_f(k+1)] P_f(k+1|k) [I - W_\infty H_f(k+1)]' \\ &\quad + W_\infty R_f(k+1) W_\infty' \end{aligned} \quad (5.6.3-1)$$

The steady-state gain  $W_\infty$  can be evaluated by solving first for the steady-state covariance  $P_\infty$  in (5.2.5-2) and then using (5.2.3-17) to find  $W_\infty$ . Since the transient characteristics may not be acceptable when a fixed steady-state gain is used, one can use a piecewise linear gain as shown in Figure 5.6.3-1. Another motivation for this is that the steady-state covariance and gain do not exist for systems with time-varying  $F(k)$ ,  $R(k)$  or  $H(k)$ .

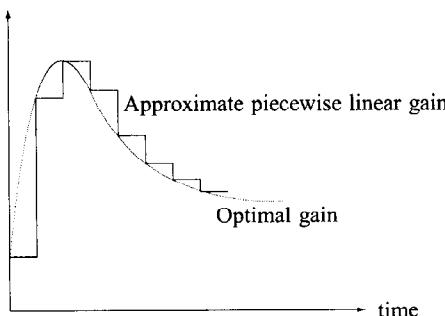


Figure 5.6.3-1: Piecewise linear approximation for filter gain.

The selection strategy for using a particular approximate state model or filter gain can be formulated as an optimization problem where the “cost” of using such approximation is minimized. For example, a suitable piecewise linear gain sequence can be chosen by minimizing the average quadratic error from the optimal gain [Gelb74].

### 5.6.4 Examples of Modeling Errors and Filter Approximations

Consider the evolution of the state  $x(k)$ , consisting of position and velocity, of a target, where the position is measured at a constant revisit interval  $T$ . The

target's state evolution is given by

$$x(k+1) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} v(k), \quad v(k) \sim \mathcal{N}(0, 1) \quad (5.6.4-1)$$

with initial state  $x(0) \sim \mathcal{N}([0 \ -10]', I)$  and the measured position is

$$z(k) = [1 \ 0] x(k) + w(k), \quad w(k) \sim \mathcal{N}(0, 1) \quad (5.6.4-2)$$

In the following examples, the effects of modeling or programming errors and implementation approximations are analyzed using DynaEst™.

### Approximate Model

The target motion is modeled as exactly a constant velocity ( $q_f = 0$ ), and a Kalman filter is implemented accordingly. In this case, the filter state equation is

$$x_f(k+1) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x_f(k) \quad (5.6.4-3)$$

Figure 5.6.4-1a shows the true and estimated states of the target under track. The actual and the filter calculated estimation variances are shown in Fig. 5.6.4-1b. It can be seen that while the filter calculated variances incorrectly converge to zero, the actual covariances continue to diverge. This is a manifestation of the mismatch between the true target state given by (5.6.4-1) and the filter model given by (5.6.4-3).

### Reduced-Order Model

The target state is estimated using a reduced-order filter. In this case, the target's velocity state component is ignored and only the position is estimated. That is, the filter model for the target state is

$$x_f(k+1) = x_f(k) + v_f(k), \quad v(k) \sim \mathcal{N}(0, q_f) \quad (5.6.4-4)$$

For this model, the state reduction matrix  $T$ , defined in (5.6.2-1), is given by  $T = [1 \ 0]$ .

Figures 5.6.4-2 and 5.6.4-3 show the state estimates and error variances (obtained using DynaEst™) for  $q_f = 1$  and  $q_f = 4$ , respectively. From these one can notice the following: (1) In both cases the actual variances do not match the filter calculated ones; (2) Increasing  $q_f$  improves the estimation results. The increased  $q_f$  serves as "artificial process noise" compensating for the reduced order model which ignores the velocity component.

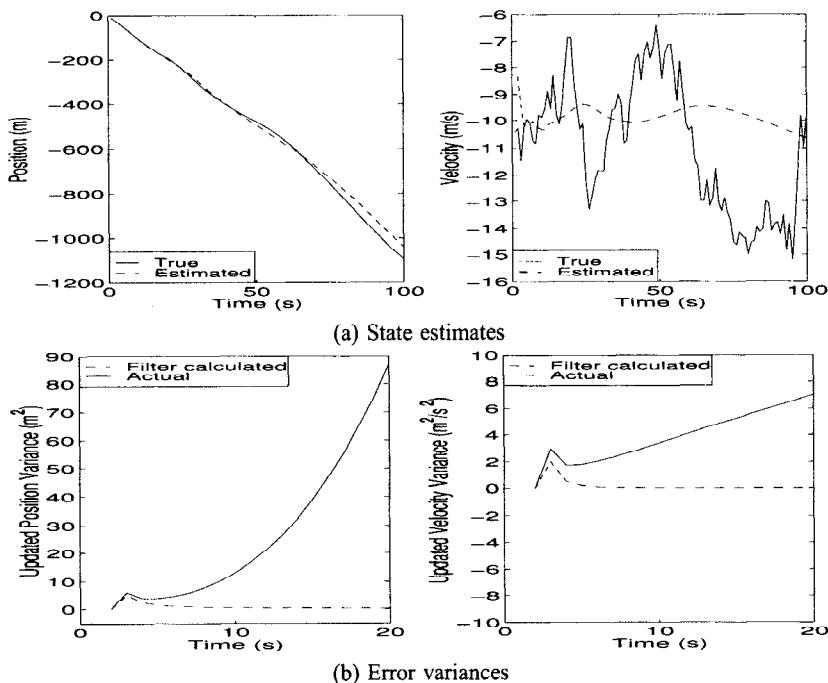
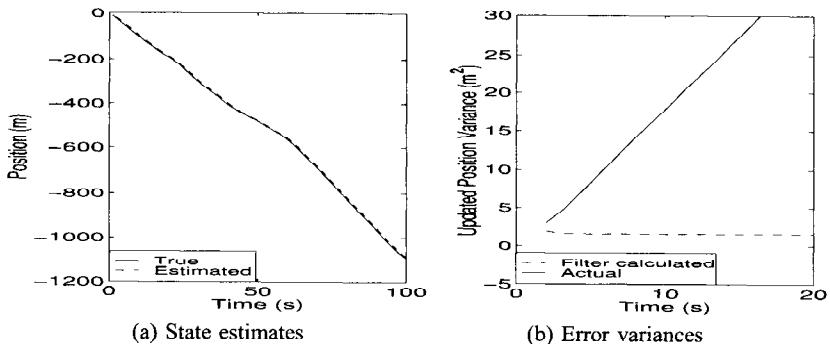


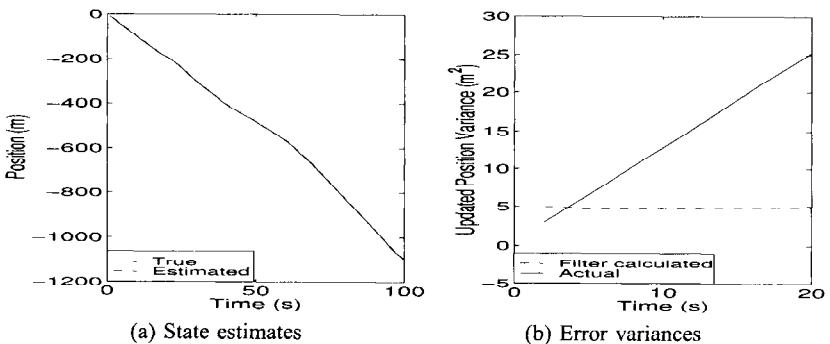
Figure 5.6.4-1: State estimates and error variances for the mismatched model with  $q_f = 0$ .

### Fixed Filter Gain

A fixed gain of  $W_f = [0.8 \ 0.5]'$  is used in place of the optimal one. The optimal and the fixed gains are shown in Fig. 5.6.4-4a. The corresponding variances are shown in Fig. 5.6.4-4b. It can be seen that the variances with incorrect gain are slightly higher than the optimal values. In other problems the difference can be larger.



**Figure 5.6.4-2:** State estimates and error variances for the reduced-order filter with  $q_f = 1$ .



**Figure 5.6.4-3:** State estimates and error variances for the reduced-order filter with  $q_f = 4$ .

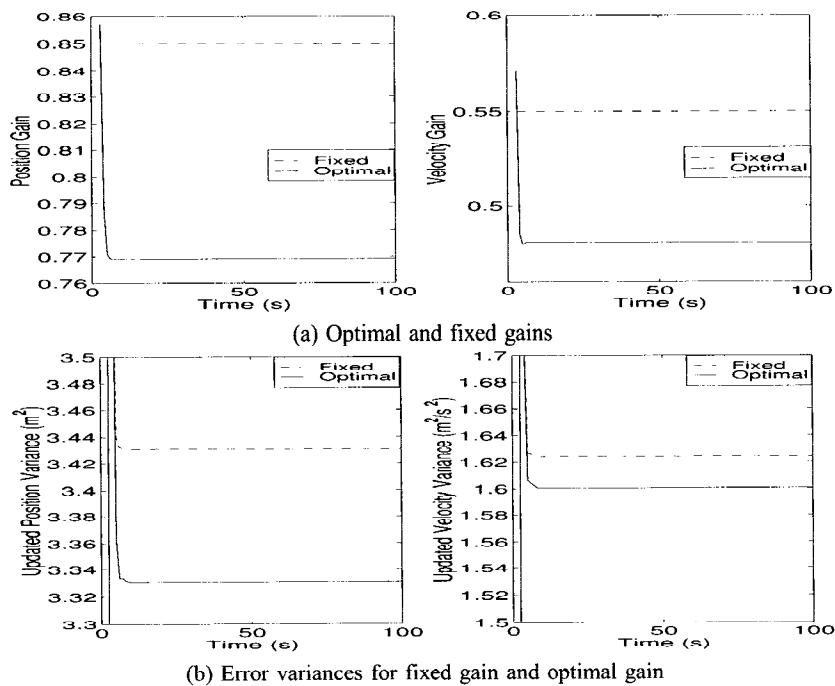


Figure 5.6.4-4: Error variances with suboptimal gains.

## 5.7 NOTES AND PROBLEMS

### 5.7.1 Bibliographical Notes

The Kalman filter originated with the work of Kalman and Bucy [Kalman60, Kalman61]. The idea of recursive estimation appeared earlier in the work of Swerling [Swerling59]. The topic of the Kalman filtering is covered in many texts, for instance, [Meditch69, Sage71, Gelb74, Anderson79, Kailath81, Maybeck79, Maybeck82, Balakrishnan84, Lewis86]. The proof of the sequential estimation for dynamic systems as a direct extension of the static case presented in Section 5.2 is simpler than the many different proofs in the literature. Issues of observability and controllability and their implications on the Kalman filter are discussed in detail in [Anderson79, Kailath81].

The concept of consistency of a state estimator (Section 5.4) has been mentioned in the literature under the name of “covariance matching.” The discussion of Section 5.4 is based on [Bar-Shalom83]. The lack of consistency of a Kalman filter has been called “divergence” (that is how severe it was in some cases) and has been the subject of extensive investigations [Sorenson85]. Sensitivity analysis of mismatched filters and the use of reduced-order models are discussed in detail in [Gelb74, Lewis86].

The technique of initialization of filters presented in Section 5.5 has been known for many years, but overlooked in many instances.

The models considered here had all known “parameters”—the matrices  $F$ ,  $G$ ,  $H$ ,  $Q$ , and  $R$ . Techniques for the “identification” of these system parameters can be found in [Ljung87].

Treatment of a constant bias in recursive filtering using state augmentation is discussed in [Friedland69].

### 5.7.2 Problems

#### 5-1 Filter implementation and consistency checking.

Consider the scalar system

$$x(k+1) = fx(k) + u(k) + v(k)$$

where  $u(k) = 1$  and  $v(k) \sim \mathcal{N}(0, q)$  and white, with measurement

$$z(k) = hx(k) + w(k)$$

where  $w(k) \sim \mathcal{N}(0, r)$  and white. The initial condition for the system is  $x(0)$ .

- Find the expression of the steady-state variance

$$P_\infty = \lim_{k \rightarrow \infty} P(k|k)$$

of the estimated state  $\hat{x}(k|k)$ .

- With the parameters of the problem  $f = h = 1$ ,  $q = 0.01$ ,  $r = 1$ , simulate a trajectory using a random number generator starting from  $x(0) = 0$  for  $k = 1, \dots, 50$ .
- Let the initial estimate of the state be  $\hat{x}(0|0) = z(0)/h$ . Determine the corresponding  $P(0|0)$  as a function of the parameters of the problem  $(h, r)$ .

4. For the values given in 2, estimate the state up to  $k = 50$ , starting as in 3. List the following:

$$k, x(k), v(k), w(k), z(k), \hat{x}(k|k-1), P(k|k-1), \tilde{z}(k|k-1), S(k), \\ \nu(k), \nu(k)/\sqrt{S(k)}, W(k), \hat{x}(k|k), P(k|k), \tilde{x}(k|k)/\sqrt{P(k|k)}$$

5. Compare the values of  $P(k|k)$  from 4 to the result of 1.
6. List  $P(k|k)$  for  $k = 0, 1, \dots, 50$  for the following values of the initial variance:  $P(0|0) = 0, 1, 10$ .
7. One desires to run the filter as in (4) with the various values of  $P(0|0)$  as in (6). What should be changed in the filter simulation?

**5-2 Random number generator testing.** Describe the tests for a “correct”  $\mathcal{N}(0, 1)$  random number generator from which we have  $n$  numbers. Indicate the distributions and probability regions for the

1. Sample mean.
2. Sample variance.
3. Sample correlation.

**5-3 Covariance of the state versus covariance of the estimation error.** Prove that, with  $\tilde{x} \stackrel{\Delta}{=} x - E[x|z]$ , one has  $\text{cov}[x|z] = \text{cov}[\tilde{x}|z]$ .

**5-4 Asymptotic distribution of the sample correlation.** Show that the sample correlation (5.4.2-12) tends to  $\mathcal{N}(0, 1/N)$ . Hint: Use the law of large numbers (where convenient) and the central limit theorem.

**5-5 MSE of the state for an update with an arbitrary gain.**

1. Prove that the Joseph form covariance update (5.2.3-18) holds for *arbitrary* gain at time  $k + 1$ . Hint: Write the propagation equation of the error from  $\tilde{x}(k+1|k)$  to  $\tilde{x}(k+1|k+1)$ .
2. State the condition for its stability.
3. Check this stability condition on the results of problem 5-1.

**5-6 Initialization of a filter (how NOT to do it — a real story).** A tracking filter was initialized in a set of Monte Carlo runs with the target’s initial range of 80 kft, initial range estimate of 100 kft and initial range variance of  $10^6 \text{ ft}^2$ .

1. Characterize the assumed quality of the initial estimate.
2. How will the average estimation error over the Monte Carlo runs behave?
3. You are a “young Turk” engineer who wants to prove mastery of estimation: Suggest a simple fix to the above initialization procedure that involves changing only the initial variance.

**5-7 Orthogonality of the innovations to the state prediction.** Show that

$$\nu(k) \perp \dot{x}(j|k-1) \quad \forall j > k-1$$

**5-8 Orthogonality of estimation error to previous estimates.** Show that

$$\tilde{x}(i|k) \perp \hat{x}(i|j) \quad \forall j \leq k$$

**5-9 Alternative derivation of the Kalman filter gain.** Show that the minimization of the trace of (5.2.3-18) with respect to the filter gain yields (5.2.3-11). *Hint:* Use the formulas from problem 1-10.

**5-10 State estimation errors' autocorrelation.** Prove that the state estimation errors are not white:

$$E[\tilde{x}(k+1|k+1)\tilde{x}(k|k)'] = [I - W(k+1)H(k+1)]F(k)P(k|k)$$

**5-11 Kalman filter with nonzero noise means.** Derive the Kalman filter equations for the formulation from Subsection 5.2.1 with the following modifications:

$$E[v(k)] = \bar{v}(k) \quad E[w(k)] = \bar{w}(k)$$

where the nonzero noise means are known. Their known covariances are  $Q$  and  $R$ . All the remaining assumptions are the same. Provide the derivations only for the equations that will be different than those in Subsection 5.2.4. Indicate which equations are not modified and why.

**5-12 Bias in the measurements.** Consider the problem from Section 5.3 with the modification that the measurement noise has an *unknown mean*  $\bar{w}$  (the sensor bias). Append this to the state as an extra component assuming it to be constant in time.

1. Indicate the new state space representation with the augmented state and *zero-mean noises* (specify the matrices  $F$ ,  $\Gamma$ , and  $H$ ).
2. What happens if we run a Kalman filter on this representation? Can one estimate this sensor bias? Justify mathematically your answer.

**5-13 Effect of the process noise on the initialization.** Modify (5.5.3-5) to account for the process noise.

**5-14 Gating.** Given a standard state estimation problem with the following specifics: the state prediction covariance matrix  $P(k+1|k) = \text{diag}(5, 16)$ , the measurement matrix  $H = I_{2 \times 2}$ , the measurement noise standard deviations as 2 and 3, respectively, and uncorrelated from each other.

1. Find the 95% point (threshold) for the NIS.
2. Given the predicted measurement  $\hat{z}(k+1|k) = [10, 15]'$  indicate if the following measurements are accepted as consistent based on the threshold from above: (a)  $z(k+1) = [7, 20]'$ ; (b)  $z(k+1) = [16, 5]'$ ; (c)  $z(k+1) = [19, 25]'$ . This procedure is called “gating” or “validation” and is used extensively to select the measurement(s) to be used in the state update when their origin is uncertain.
3. Repeat the above two items for the 99% threshold.

- 5-15 Three-point initialization.** Consider three position measurements  $z(i)$ ,  $i = 1, 2, 3$ , taken at fixed intervals  $T$ . Assuming a constant acceleration motion, find

1. The LS estimates of the position, velocity and acceleration at  $i = 3$
2. The corresponding covariance matrix.

- 5-16 Fusion of estimates.** Consider the two state estimates (vectors)

$$z_i \stackrel{\Delta}{=} \hat{x}_i = x + \tilde{x}_i \quad i = 1, 2$$

with the estimation errors  $\tilde{x}_i$  zero-mean, with covariances

$$E[\tilde{x}_i \tilde{x}'_j] = P_{ij} \quad i, j = 1, 2$$

find the MLE  $\hat{x}(z_1, z_2)$  and its covariance assuming the above errors are jointly Gaussian with the above moments.

- 5-17 Sensitivity to process noise variance.** Consider a first-order dynamic system with  $F = 1$ ,  $H = 1$ , process noise variance  $q$  and measurement noise variance  $r$ .

1. Find the optimal filter gain in steady state,  $W_\infty$ , as a function of  $q$  and  $r$ .
2. Show that it can be written as a function of  $\mu = q/r$  only.
3. Assume that a fixed-gain nonoptimal filter is used to estimate  $x$ , based on the (incorrect) assumption that the process noise variance is  $q_f$ . Give the expression of this gain,  $W_{\infty_f}$ . Indicate the equations to be used to evaluate the resulting estimation MSE for  $x$ . Solve for the steady-state value  $P_\infty(q, r, q_f)$  of this MSE.
4. Indicate the condition on  $W_{\infty_f}$  for the existence of the steady-state value  $P_\infty(q, r, q_f)$ .
5. Plot  $P_\infty(q, r, q_f)$  for  $q = 1$ ,  $r = 1$  for  $q_f \in [10^{-2}, 10^2]$ .

- 5-18 Simultaneous estimation of sensor bias and target location.** A sensor is believed to be at the origin of the real axis. However, its true location is biased and known only to the following extent: It is on the real axis in the neighborhood of the origin at  $b(k)$ , which is a Gaussian zero-mean random sequence with autocorrelation

$$E[b(k)b(j)] = \frac{\sigma_v^2}{1 - \alpha^2} \alpha^{|k-j|}$$

This sensor measures the range (distance) to a fixed *unknown* location point on the real axis at  $x_1$  in the presence of additive Gaussian zero-mean white noise with s.d.  $\sigma_r$ .

- (i) Formulate the estimation problem for the above using the appropriate state space representation.
- (ii) Is the system completely observable? What condition has to be met?
- (iii) Find the initial-state covariance matrix based on the above.
- (iv) Find the steady-state estimation covariance matrix.

### 5.7.3 Computer Applications

The computational parts of these computer application exercises can be solved using the companion software DynaEst<sup>TM</sup>.

- 5-1 Evaluation of a Kalman filter with different sampling intervals.** Consider the single-input, single-output linear system with transfer function

$$H(s) = \frac{Z_1(s)}{V(s)} = \frac{1}{s^2}$$

driven by a white noise  $\tilde{v}(t)$  with power spectral density  $\tilde{q}$ .

1. Find its continuous-time state-space representation with states (position and velocity)

$$x_1 = z_1 \quad x_2 = \dot{z}_1$$

2. Assume the output (position) is measured every  $T$  seconds

$$z(k) = z_1(k) = x_1(k) + w(k)$$

in the presence of a zero mean white noise with variance  $R_1$ . Find the discrete-time state representation and evaluate the process noise covariance matrix.

3. For values  $x_1(0) = 0\text{ m}$ ,  $x_2(0) = 10\text{ m/s}$ ,  $T = 2\text{ s}$ ,  $\tilde{q} = 1\text{ m}^2/\text{s}^3$ ,  $R_1 = 9\text{ m}^2$ , initialize a (discrete-time) KF for this system with two-point differencing and run one sample trajectory for 100 s. Plot the following:

- (i) True position, the measurements and the estimated position
  - (ii) True velocity and the estimated velocity
  - (iii) The estimated position standard deviation
  - (iv) The estimated velocity standard deviation.
4. Redo the filter for  $T = 5\text{ s}$ .

- 5-2 Evaluation of a Kalman filter with different measurements.** For the same system as in computer application exercise 5-1, assume the output measured every  $T$  seconds is

$$z(k) = \begin{bmatrix} z_1(k) \\ z_2(k) \end{bmatrix} = x(k) + w(k)$$

in the presence of a zero mean white noise with covariance

$$R = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix}$$

Redo item 3 from computer application 5-1 for  $r_{11} = 9\text{ m}^2$ ,  $r_{22} = 9\text{ m}^2/\text{s}^4$  with correlation coefficient  $\rho$  between the position and velocity measurements

1.  $\rho = 0$
2.  $\rho = 0.5$
3.  $\rho = -0.5$

Order these three configurations according to the resulting estimation quality and explain why they turned out as they did.

**5-3 Observability of a linear dynamic system.** Consider the following linear dynamic systems with

(a)

$$F_a = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, Q_a = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, H_a = [1 \quad -1], R_a = 1 \text{ and } P_a(0|0) = \begin{bmatrix} 1 & 0 \\ 0 & 100 \end{bmatrix}$$

(b)

$$F_b = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & \alpha \end{bmatrix}, Q_b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \alpha = 0.5, H_b = [1 \quad -1 \quad 0], R_b = 1 \text{ and}$$

$$P_b(0|0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

1. Using DynaEst, list  $P_a(k|k)$  and  $P_b(k|k)$  for  $k = 0, 1, 2, 5, 10, 20, 50, 100$ .
2. Explain the results.
3. What range of values of  $\alpha$  is “good?” Justify.

# Chapter 6

## ESTIMATION FOR KINEMATIC MODELS

### 6.1 INTRODUCTION

#### 6.1.1 Outline

This chapter discusses a class of widely used models derived from simple equations of motion — constant velocity and constant acceleration. These models are more general in the sense that the corresponding (second- and third-order) derivatives of the position are not zero, but a zero-mean random process.

Section 6.2 presents the discrete-time kinematic model obtained by discretizing the continuous-time state space representation driven by white noise. The state model defined directly in discrete time using a piecewise constant white random sequence as process noise is presented in Section 6.3.

Section 6.4 presents explicit filters for noiseless kinematic models.

For noisy kinematic models, explicit steady-state filters are derived in Section 6.5. These filters corresponding to second- and third-order models are known as the  $\alpha\text{-}\beta$  and  $\alpha\text{-}\beta\text{-}\gamma$  filters, respectively, and their gains are expressed in terms of the **target maneuvering index** — the ratio of the motion and the observation uncertainties. Since the statistical characterization of the process noise is a key **filter design parameter**, this is discussed in detail. Subsequently, one important aspect of tracking **system design** — selection of the **sampling frequency**, i.e., the **revisit rate** — is discussed.

The models are presented for a single coordinate. For motion in several coordinates, it is customary to use such models assumed independent across coordinates — this leads to “decoupled” filtering. Finally, some intuitive insight into filter design is presented.

#### 6.1.2 Kinematic Models — Summary of Objectives

Define the following kinematic models

- White noise acceleration (second-order model)
- Wiener process acceleration (third-order model)

Derive discrete-time kinematic models by

- Discretizing the continuous-time state space representation driven by white noise
- Directly defining the state model in discrete time using a piecewise-constant white random sequence as process noise

Present

- Explicit filters for noiseless kinematic models
- Explicit steady-state filters for noisy kinematic models
  - $\alpha\text{-}\beta$
  - $\alpha\text{-}\beta\text{-}\gamma$

with their gains expressed in terms of the target maneuvering index.

Discuss

- Filter design and “noise reduction”
- System design — selection of the sampling frequency (revisit rate)

## 6.2 DISCRETIZED CONTINUOUS-TIME KINEMATIC MODELS

### 6.2.1 The Kinematic Models

**Kinematic state models** are defined by setting a certain derivative of the position to zero. In the absence of any random input they yield motion characterized by a polynomial in time. Such models are also called **polynomial models**, and the corresponding state estimation filters are sometimes referred to as **polynomial filters**.

Since it is not realistic to assume that there are no disturbances, one can model them as random inputs. One way of modeling this is via a *continuous-time white process noise*.

Since, in general, the state observations are done in discrete time, the corresponding discrete-time state equations are needed. Subsection 6.2.2 presents the **white noise acceleration** state model, which is two-dimensional per coordinate. The **Wiener process acceleration** state model, which is three-dimensional per coordinate, is presented in Subsection 6.2.3.

In many applications the same model is used for each coordinate. In some applications — for instance, in air traffic control — one can use two third-order models for the horizontal motion and a second-order model for the (more benign) vertical motion.

In general the motion along each coordinate is assumed “decoupled” from the other coordinates. The noises entering into the various coordinates are also assumed to be mutually independent with possibly different variances. The discussion in this section will deal with kinematic models in *one generic coordinate*.

### 6.2.2 Continuous White Noise Acceleration Model

A *constant velocity object* moving in a generic coordinate  $\xi$  is described by the equation

$$\ddot{\xi}(t) = 0 \quad (6.2.2-1)$$

Since the position  $\xi(t)$  evolves, in the absence of noise, according to a polynomial in time (in this case, of second order), this model is also called a *polynomial model*.

In practice, the velocity undergoes at least slight changes. This can be modeled by a continuous time zero-mean white noise  $\tilde{v}$  as

$$\ddot{\xi}(t) = \tilde{v}(t) \quad (6.2.2-2)$$

where

$$E[\tilde{v}(t)] = 0 \quad (6.2.2-3)$$

$$E[\tilde{v}(t)\tilde{v}(\tau)] = \tilde{q}(t)\delta(t - \tau) \quad (6.2.2-4)$$

The continuous-time process noise intensity  $\tilde{q}$ , which is (when time-invariant) its power spectral density, is, in general, a design parameter for the estimation filter based on this model. This will be discussed later in more detail in Section 6.5.

The state vector corresponding to (6.2.2-2), which is two-dimensional per coordinate, is

$$x = [\xi \quad \dot{\xi}]^T \quad (6.2.2-5)$$

Thus, this model will be called the *continuous white noise acceleration (CWNA) model* or *second-order kinematic model* (double integrator). Note that the velocity in this model is a Wiener process — the integral of white noise.

The continuous-time state equation is

$$\dot{x}(t) = Ax(t) + D\tilde{v}(t) \quad (6.2.2-6)$$

where

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (6.2.2-7)$$

$$D = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (6.2.2-8)$$

## The Discretized State Equation

The discrete-time state equation with sampling period  $T$  is

$$x(k+1) = Fx(k) + v(k) \quad (6.2.2-9)$$

where (see Subsection 4.3.1)

$$F = e^{AT} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad (6.2.2-10)$$

and the discrete-time process noise relates to the continuous time one as

$$v(k) = \int_0^T e^{A(T-\tau)} D\tilde{v}(kT + \tau) d\tau \quad (6.2.2-11)$$

From the above, the covariance of the discrete-time process noise  $v(k)$ , assuming  $\tilde{q}$  to be constant and using (6.2.2-4), is

$$\begin{aligned} Q &= E[v(k)v(k)'] = \int_0^T \begin{bmatrix} T - \tau \\ 1 \end{bmatrix} [T - \tau \quad 1] \tilde{q} d\tau \\ &= \begin{bmatrix} \frac{1}{3}T^3 & \frac{1}{2}T^2 \\ \frac{1}{2}T^2 & T \end{bmatrix} \tilde{q} \end{aligned} \quad (6.2.2-12)$$

### Guideline for Choice of Process Noise Intensity

The changes in the velocity over a sampling period  $T$  are of the order of

$$\sqrt{Q_{22}} = \sqrt{\tilde{q}T} \quad (6.2.2-13)$$

This can serve as a guideline for **process noise intensity choice** — the choice of the power spectral density  $\tilde{q}$  of the process noise in this model.

Note that the physical dimension of  $\tilde{q}$  is [length]<sup>2</sup>/[time]<sup>3</sup>.

A **nearly constant velocity (NCV) model**<sup>1</sup> is obtained by the choice of a “small” intensity  $\tilde{q}$  in the following sense: The changes in the velocity have to be small compared to the actual velocity.

### 6.2.3 Continuous Wiener Process Acceleration Model

The motion of a constant acceleration object for a generic coordinate  $\xi$  is described by the equation

$$\ddot{\xi}(t) = 0 \quad (6.2.3-1)$$

---

<sup>1</sup>The term **constant velocity (CV) model** is used sometimes in the literature, with some abuse of language, for this model regardless of the intensity of the process noise.

Similarly to (6.2.2-2), the acceleration is not exactly constant and its changes can be modeled by a continuous-time zero-mean white noise as

$$\ddot{\xi}(t) = \tilde{v}(t) \quad (6.2.3-2)$$

Note that in this case the acceleration is a Wiener process — hence the name ***continuous Wiener process acceleration (CWPA) model***. Since the derivative of the acceleration is the jerk, this model can also be called the ***white noise jerk model***.<sup>2</sup>

The state vector corresponding to the above is

$$x = [\xi \quad \dot{\xi} \quad \ddot{\xi}]' \quad (6.2.3-3)$$

and its continuous-time state equation is

$$\dot{x}(t) = Ax(t) + D\tilde{v}(t) \quad (6.2.3-4)$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (6.2.3-5)$$

This is a ***third-order model*** with three integrations: All three eigenvalues of  $A$  — the poles of the continuous time transfer function — are zero.

### Remark

One can have other third-order models, for instance, with the acceleration having an exponentially decaying autocorrelation, rather than being a Wiener process. The ***exponentially autocorrelated acceleration model*** is presented later in Subsection 8.2.2.

### The Discretized State Equation

The discrete-time state equation with sampling period  $T$  is

$$x(k+1) = Fx(k) + v(k) \quad (6.2.3-6)$$

with the transition matrix

$$F = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \quad (6.2.3-7)$$

---

<sup>2</sup>In view of this, one has the mathematical epithet “x triple dot.”

and the covariance matrix of  $v(k)$  given by

$$Q = E[v(k)v(k)'] = \begin{bmatrix} \frac{1}{20}T^5 & \frac{1}{8}T^4 & \frac{1}{6}T^3 \\ \frac{1}{8}T^4 & \frac{1}{3}T^3 & \frac{1}{2}T^2 \\ \frac{1}{6}T^3 & \frac{1}{2}T^2 & T \end{bmatrix} \tilde{q} \quad (6.2.3-8)$$

Note the three unity eigenvalues of the transition matrix  $F$  in (6.2.3-7) — the poles of the discrete-time transfer function — corresponding to the three integrations.

### Guideline for Choice of Process Noise Intensity

The changes in the acceleration over a sampling period  $T$  are of the order of

$$\sqrt{Q_{33}} = \sqrt{\tilde{q}T} \quad (6.2.3-9)$$

This can serve as a guideline in the **process noise intensity choice** — the choice of the power spectral density  $\tilde{q}$  of the continuous-time process noise  $\tilde{v}$  for “tuning” this model to the actual motion of the object of interest.

A **nearly constant acceleration (NCA) model**<sup>3</sup> is obtained by choosing a “small” intensity  $\tilde{q}$  in the following sense: The changes in the acceleration, which are of the order of  $\sqrt{\tilde{q}T}$  over an interval  $T$ , should be small relative to the actual acceleration levels.

## 6.3 DIRECT DISCRETE-TIME KINEMATIC MODELS

### 6.3.1 Introduction

The discrete time plant equation for the continuous time white noise acceleration and white noise jerk (Wiener process acceleration) were given in the previous section.

Another common kinematic model is directly defined in discrete time as follows. The discrete-time process noise  $v(k)$  is a scalar-valued *zero-mean white sequence*

$$E[v(k)v(j)] = \sigma_v^2 \delta_{kj} \quad (6.3.1-1)$$

and enters into the dynamic equation as follows:

$$x(k+1) = Fx(k) + \Gamma v(k) \quad (6.3.1-2)$$

where the **noise gain**  $\Gamma$  is an  $n_x$ -dimensional vector.

---

<sup>3</sup>The term **constant acceleration (CA) model** is used sometimes in the literature, with some abuse of language, for this model regardless of the intensity of the process noise.

The assumption in the second-order model is that the object undergoes a *constant acceleration* during each sampling period (of length  $T$ )

$$\tilde{v}(t) = v(k) \quad t \in [kT, (k+1)T] \quad (6.3.1-3)$$

and that these accelerations are *uncorrelated from period to period*. The above indicates a ***piecewise constant acceleration***.

### Remark

It is clear that if the above assumption is correct for a given sampling period  $T_1$ , then it cannot be correct for any other  $T_2$  (except integer multiples of  $T_1$ ). Neither this *piecewise constant white noise* assumption nor the *continuous-time white noise* (6.2.2-4) are completely realistic — both are approximations. Nevertheless, with judicious choices of their intensities, both can successfully model motion uncertainties over short to moderate intervals of time.

### 6.3.2 Discrete White Noise Acceleration Model

If  $v(k)$  is the *constant acceleration* during the  $k$ th sampling period (of length  $T$ ), the increment in the velocity during this period is  $v(k)T$ , while the effect of this acceleration on the position is  $v(k)T^2/2$ .

The state equation for the ***piecewise constant white acceleration model***, or ***discrete white noise acceleration (DWNA) model***, which is of second order, is

$$x(k+1) = Fx(k) + \Gamma v(k) \quad (6.3.2-1)$$

with the process noise  $v(k)$  a *zero-mean white acceleration sequence*.

The transition matrix is

$$F = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad (6.3.2-2)$$

and the vector gain multiplying the scalar process noise is given, in view of the above discussion, by

$$\Gamma = \begin{bmatrix} \frac{1}{2}T^2 \\ T \end{bmatrix} \quad (6.3.2-3)$$

The covariance of the process noise multiplied by the gain,  $\Gamma v(k)$ , is

$$Q = E[\Gamma v(k)v(k)\Gamma'] = \Gamma\sigma_v^2\Gamma' = \begin{bmatrix} \frac{1}{4}T^4 & \frac{1}{2}T^3 \\ \frac{1}{2}T^3 & T^2 \end{bmatrix}\sigma_v^2 \quad (6.3.2-4)$$

Note the difference between this and (6.2.2-12).

The physical dimension of  $v$  and  $\sigma_v$  is [length]/[time]<sup>2</sup>, i.e., that of acceleration.

## Guideline for Choice of Process Noise Variance

For this model,  $\sigma_v$  should be of the order of the maximum acceleration magnitude  $a_M$ . A practical range is  $0.5a_M \leq \sigma_v \leq a_M$ .

A **nearly constant velocity (NCV) model**<sup>4</sup> is obtained by the choice of a “small” intensity  $q$  in the following sense: The changes in the velocity over a sampling interval, which are of the order of  $\sigma_v T$ , have to be small compared to the actual velocity.

## Note on the Multidimensional Case

When motion is in several coordinates, then, with decoupled filtering across coordinates, (6.3.2-4) is a block of the overall  $Q$ , which is then block diagonal.

### 6.3.3 Discrete Wiener Process Acceleration Model

For the **piecewise constant Wiener process acceleration model**, or **discrete Wiener process acceleration (DWPA) model**, the (third order) state equation is

$$x(k+1) = Fx(k) + \Gamma v(k) \quad (6.3.3-1)$$

where

$$F = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \quad (6.3.3-2)$$

$$\Gamma = \begin{bmatrix} \frac{1}{2}T^2 \\ T \\ 1 \end{bmatrix} \quad (6.3.3-3)$$

In this model, the white process noise  $v(k)$  is the **acceleration increment** during the  $k$ th sampling period and it is assumed to be a **zero-mean white sequence** — the **acceleration is a discrete-time Wiener process**. The formulation in terms of acceleration increment is more convenient than the one in terms of the third-order derivative (jerk).

The covariance of the process noise multiplied by the gain  $\Gamma$  is

$$Q = \Gamma \sigma_v^2 \Gamma' = \begin{bmatrix} \frac{1}{4}T^4 & \frac{1}{2}T^3 & \frac{1}{2}T^2 \\ \frac{1}{2}T^3 & T^2 & T \\ \frac{1}{2}T^2 & T & 1 \end{bmatrix} \sigma_v^2 \quad (6.3.3-4)$$

<sup>4</sup>As in the continuous-time case, the term **constant velocity (CV) model** is used sometimes in the literature, with some abuse of language, for this model regardless of the intensity of the process noise.

## Guideline for Choice of Process Noise Variance

For this model,  $\sigma_v$  should be of the order of the magnitude of the maximum acceleration increment over a sampling period,  $\Delta a_M$ . A practical range is  $0.5\Delta a_M \leq \sigma_v \leq \Delta a_M$ .

## The Process Noise Variance for Different Sampling Periods

Note that if the sampling period is changed, one has to carry out a *rescaling of the variance of the process noise*. (See problems 6-1 through 6-4.)

### 6.3.4 Kinematic Models — Summary

Kinematic (polynomial) model of order  $n$ : The  $n$ th derivative of the position is equal to

- Zero — noiseless model
- White noise — noisy model

There are two major classes of noisy discrete time kinematic models:

1. Obtained from discretization of the continuous-time model, driven by *continuous-time white noise*, for a given sampling period.
2. Obtained by direct definition of the process noise in discrete time as a *piecewise constant white sequence* — the process noise is assumed to be constant over each sampling period and independent between periods.

Within each class the following models were discussed in detail:

- White noise acceleration (WNA) — second-order model (sometimes called CV)
- Wiener process acceleration (WPA) — third-order model (sometimes called CA)

The resulting process noise covariance matrices for the two classes of models are different in their dependence on the sampling period.

The process noise covariance matrices in the direct discrete time models are positive semidefinite of rank 1, while their counterparts from the discretized continuous time models are of full rank.

Both models are, obviously, approximations.

The more commonly used model is the one in item 2. Its advantage is that the process noise intensity in this case is easily related to physical characteristics of the motion (acceleration). The model from item 1 is more convenient when one deals with variable sampling intervals.

## 6.4 EXPLICIT FILTERS FOR NOISELESS KINEMATIC MODELS

### 6.4.1 LS Estimation for Noiseless Kinematic Models

Consider an object moving with constant velocity (without process noise). Such a case was treated in Subsection 3.5.1, where, using least squares, the estimate of the *initial position and the (constant) velocity* were obtained, both in batch and in recursive form.

Using these results, the recursion for the estimate of the *current state*  $x(k) \triangleq x(t_k)$  of a second-order **noiseless kinematic model** — the (exact) **constant velocity (CV) model** — will be obtained.

Similarly to (6.2.2-9), but without the process noise, one has

$$x(t) = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} x(0) \quad (6.4.1-1)$$

Assuming a uniform sampling rate with sampling period  $T$ , we obtain

$$\hat{x}(t_{k+1}|t_{k+1}) \triangleq \hat{x}(k+1|k+1) = \begin{bmatrix} 1 & (k+1)T \\ 0 & 1 \end{bmatrix} \hat{x}(0|k+1) \quad (6.4.1-2)$$

In the above equation  $\hat{x}(0|k+1)$  is the estimate of the *initial state*, with the initial time 0, to which it pertains, now explicitly indicated. This estimate of the initial state was obtained via recursive LS in (3.5.1-8), where it was denoted, without the first time index, as  $\hat{x}(k+1)$ .

### 6.4.2 The KF for Noiseless Kinematic Models

The recursion for the estimate of the current state is

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + W(k+1)[z(k+1) - \hat{z}(k+1|k)] \quad (6.4.2-1)$$

where

$$\hat{x}(k+1|k) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \hat{x}(k|k) \quad (6.4.2-2)$$

$$\hat{z}(k+1|k) = [1 \ 0] \hat{x}(k+1|k) \quad (6.4.2-3)$$

Note that (6.4.2-1) is actually a recursive LS estimator “disguised” as a Kalman filter. In view of this, the gain  $W(k)$  in (6.4.2-1) can be obtained directly from the gain (3.5.1-18), denoted now as  $W_0(k)$  (since it pertains to the initial state), by multiplying it with the same matrix as in (6.4.1-2)

$$\begin{aligned} W(k+1) &= \begin{bmatrix} 1 & (k+1)T \\ 0 & 1 \end{bmatrix} W_0(k+1) \\ &= \begin{bmatrix} 1 & (k+1)T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -\frac{2}{(k+1)} \\ \frac{6}{(k+1)(k+2)T} \end{bmatrix} = \begin{bmatrix} \frac{4k+2}{(k+1)(k+2)} \\ \frac{6}{(k+1)(k+2)T} \end{bmatrix} \end{aligned} \quad (6.4.2-4)$$

It can be easily shown that the covariance associated with the estimate (6.4.1-2) is, using (3.5.1-19), given by

$$\begin{aligned} P(k+1|k+1) &= \begin{bmatrix} 1 & (k+1)T \\ 0 & 1 \end{bmatrix} P(k+1) \begin{bmatrix} 1 & (k+1)T \\ 0 & 1 \end{bmatrix}' \\ &= \frac{2\sigma^2}{(k+1)(k+2)} \begin{bmatrix} 2k+1 & \frac{3}{T} \\ \frac{3}{T} & \frac{6}{kT^2} \end{bmatrix} \end{aligned} \quad (6.4.2-5)$$

## Summary

For noiseless kinematic (polynomial) models, one can obtain explicit expressions of the Kalman filter gain and the state covariance.

Similar results can be obtained for the (exact) **constant acceleration (CA) model**.

Due to the absence of process noise, the state covariance converges to zero and so does the filter gain.

## 6.5 STEADY-STATE FILTERS FOR NOISY KINEMATIC MODELS

### 6.5.1 The Problem

As indicated in Subsection 5.2.5, the state estimation covariance for a time-invariant system (with constant coefficients in the state and measurement equations) will converge under suitable conditions to a steady-state value.

These conditions are satisfied for the kinematic models described in Sections 6.2 and 6.3. Furthermore, *explicit expressions of the steady-state covariance and filter gain* can be obtained.

It will be assumed that only position measurements are available, that is,

$$z(k) = Hx(k) + w(k) \quad (6.5.1-1)$$

where for the white noise acceleration (second-order) model

$$H = [1 \ 0] \quad (6.5.1-2)$$

and for the Wiener process acceleration (third-order) model

$$H = [1 \ 0 \ 0] \quad (6.5.1-3)$$

The measurement noise autocorrelation function is

$$E[w(k)w(j)] = R\delta_{kj} = \sigma_w^2 \delta_{kj} \quad (6.5.1-4)$$

The resulting *steady-state filters for noisy kinematic models* are known as *alpha-beta* and *alpha-beta-gamma* filters<sup>5</sup> for the second and third-order models, respectively. The coefficients  $\alpha$ ,  $\beta$ , and  $\gamma$  yield the filter steady-state gain vector components.

Subsection 6.5.2 presents the methodology of the derivation of the  $\alpha\text{-}\beta$  filter. This is used for the direct discrete time and the discretized continuous-time second-order models in Subsections 6.5.3 and 6.5.4, respectively. Subsection 6.5.5 presents the  $\alpha\text{-}\beta\text{-}\gamma$  filter for the direct discrete time third-order model.

### 6.5.2 Derivation Methodology for the Alpha-Beta Filter

The steady-state filter for the two-dimensional kinematic model is obtained as follows. The plant equation is

$$x(k+1) = Fx(k) + v(k) \quad (6.5.2-1)$$

where

$$F = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad (6.5.2-2)$$

and the (vector-valued) process noise has the autocorrelation function

$$E[v(k)v(j)'] = Q\delta_{kj} \quad (6.5.2-3)$$

The measurement is given by (6.5.1-1), (6.5.1-2), and (6.5.1-4). The variance of the (scalar) measurement noise will be denoted as  $\sigma_w^2 \triangleq R$ .

The steady-state values of the components of the state estimation covariance matrix will be denoted as

$$\lim_{k \rightarrow \infty} P(k|k) = [p_{ij}] \quad (6.5.2-4)$$

The components of the one-step prediction covariance are denoted as

$$\lim_{k \rightarrow \infty} P(k+1|k) = [m_{ij}] \quad (6.5.2-5)$$

while for the *alpha-beta filter gain* the notation will be

$$\lim_{k \rightarrow \infty} W(k) \triangleq [g_1 \ g_2]' \triangleq \left[ \alpha \ \frac{\beta}{T} \right]'$$

(6.5.2-6)

Note that, as defined,  $\alpha$  and  $\beta$  are dimensionless.

---

<sup>5</sup>Also called f-g and f-g-h filters in the literature [Brookner98].

## Note

The existence, uniqueness and positive definiteness of (6.5.2-4) are guaranteed since the required observability and controllability conditions are satisfied (see problem 6-6).

The expression of the innovation covariance (5.2.3-9) yields

$$S = H \begin{bmatrix} m_{11} & m_{12} \\ m_{12} & m_{22} \end{bmatrix} H' + R = m_{11} + \sigma_w^2 \quad (6.5.2-7)$$

where the notation  $\sigma_w^2 = R$  is now used.

The filter gain given by (5.2.3-11) becomes

$$W = \begin{bmatrix} m_{11} & m_{12} \\ m_{12} & m_{22} \end{bmatrix} H' S^{-1} = \begin{bmatrix} \frac{m_{11}}{m_{11} + \sigma_w^2} & \frac{m_{12}}{m_{11} + \sigma_w^2} \end{bmatrix}' \quad (6.5.2-8)$$

From (6.5.2-6) and (6.5.2-8) it follows that

$$g_1 = \frac{m_{11}}{m_{11} + \sigma_w^2} \quad (6.5.2-9)$$

$$g_2 = \frac{m_{12}}{m_{11} + \sigma_w^2} = g_1 \frac{m_{12}}{m_{11}} \quad (6.5.2-10)$$

The covariance update (5.2.3-15) becomes, using (6.5.2-8) to (6.5.2-10),

$$\begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} = (I - WH) \begin{bmatrix} m_{11} & m_{12} \\ m_{12} & m_{22} \end{bmatrix} = \begin{bmatrix} (1 - g_1)m_{11} & (1 - g_1)m_{12} \\ (1 - g_1)m_{12} & m_{22} - g_2 m_{12} \end{bmatrix} \quad (6.5.2-11)$$

The covariance prediction equation (5.2.3-5) is rewritten as follows:

$$P(k|k) = F^{-1}[P(k+1|k) - Q](F^{-1})' \quad (6.5.2-12)$$

where, from (6.5.2-2), one has

$$F^{-1} = \begin{bmatrix} 1 & -T \\ 0 & 1 \end{bmatrix} \quad (6.5.2-13)$$

The steady-state solution for the covariance and gains is obtained from the set of nonlinear equations (6.5.2-9) to (6.5.2-12) using the suitable expression of the process noise covariance  $Q$  in (6.5.2-12). The expression for the direct discrete time model is (6.3.2-4), while for the discretized continuous-time model it is (6.2.2-12).

### 6.5.3 The Alpha-Beta Filter for the DWNA Model

Using the process noise covariance (6.3.2-4), which corresponds to a *discrete white noise acceleration (DWNA) model* in the form of a *piecewise constant white process noise* — accelerations that are constant over each sampling period and uncorrelated from period to period — yields in (6.5.2-12)

$$\begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} = \begin{bmatrix} m_{11} - 2Tm_{12} + T^2m_{22} - \frac{1}{4}T^4\sigma_v^2 & m_{12} - Tm_{22} + \frac{1}{2}T^3\sigma_v^2 \\ m_{12} - Tm_{22} + \frac{1}{2}T^3\sigma_v^2 & m_{22} - T^2\sigma_v^2 \end{bmatrix} \quad (6.5.3-1)$$

Equating the terms of (6.5.2-11) and (6.5.3-1) yields, after some cancellations,

$$g_1 m_{11} = 2Tm_{12} - T^2m_{22} + \frac{T^4}{4}\sigma_v^2 \quad (6.5.3-2)$$

$$g_1 m_{12} = Tm_{22} - \frac{T^3}{2}\sigma_v^2 \quad (6.5.3-3)$$

$$g_2 m_{12} = T^2\sigma_v^2 \quad (6.5.3-4)$$

Equations (6.5.2-9), (6.5.2-10), and (6.5.3-2) to (6.5.3-4) with the five unknowns  $g_1$ ,  $g_2$ ,  $m_{11}$ ,  $m_{12}$ , and  $m_{22}$  are solved next.

From (6.5.2-9) and (6.5.2-10) one has

$$m_{11} = \frac{g_1}{1 - g_1}\sigma_w^2 \quad (6.5.3-5)$$

$$m_{12} = \frac{g_2}{1 - g_1}\sigma_w^2 \quad (6.5.3-6)$$

From (6.5.3-3) and (6.5.3-4) one obtains

$$m_{22} = \frac{g_1 m_{12}}{T} + \frac{T^2}{2}\sigma_v^2 = \left(\frac{g_1}{T} + \frac{g_2}{2}\right)m_{12} \quad (6.5.3-7)$$

Using (6.5.3-4) to (6.5.3-7) in (6.5.3-2) yields

$$\frac{g_1^2}{1 - g_1}\sigma_w^2 = 2T\frac{g_2}{1 - g_1}\sigma_w^2 - T^2\left(\frac{g_1}{T} + \frac{g_2}{2}\right)\frac{g_2}{1 - g_1}\sigma_w^2 + \frac{T^2}{4}\frac{g_2^2}{1 - g_1}\sigma_w^2 \quad (6.5.3-8)$$

which, after cancellations, becomes

$$g_1^2 - 2Tg_2 + Tg_1g_2 + \frac{T^2}{4}g_2^2 = 0 \quad (6.5.3-9)$$

With the dimensionless variables  $\alpha$  and  $\beta$  one has

$$\alpha^2 - 2\beta + \alpha\beta + \frac{\beta^2}{4} = 0 \quad (6.5.3-10)$$

which yields the first equation for  $\alpha$  and  $\beta$  as

$$\alpha = \sqrt{2\beta} - \frac{\beta}{2} \quad (6.5.3-11)$$

The second equation for  $\alpha$  and  $\beta$  follows immediately from (6.5.3-4) and (6.5.3-6) as

$$m_{12} = \frac{T^2 \sigma_v^2}{\beta/T} = \frac{\beta/T}{1-\alpha} \sigma_w^2 \quad (6.5.3-12)$$

or

$$\frac{\beta^2}{1-\alpha} = \frac{T^4 \sigma_v^2}{\sigma_w^2} \triangleq \lambda^2 \quad (6.5.3-13)$$

The quantity

$$\boxed{\lambda \triangleq \frac{\sigma_v T^2}{\sigma_w}} \quad (6.5.3-14)$$

is called the *target maneuvering index* (also called *target tracking index*) since it is proportional to the ratio of

- The ***motion uncertainty*** — the RMS value of the process noise (acceleration) effect on the position over one period, which is  $\sigma_v T^2/2$  — see (6.3.2-4).
- The ***observation uncertainty*** — the measurement noise RMS value  $\sigma_w$ .

Eliminating  $\alpha$  from (6.5.3-13) with (6.5.3-11) yields

$$\frac{\beta^2}{1-\alpha} = \frac{\beta^2}{1 - \sqrt{2\beta} + \beta/2} = \frac{\beta^2}{(1 - \sqrt{\beta/2})^2} = \lambda^2 \quad (6.5.3-15)$$

or

$$\beta + \frac{\lambda}{\sqrt{2}} \sqrt{\beta} - \lambda = 0 \quad (6.5.3-16)$$

The positive solution for  $\sqrt{\beta}$  from the above is

$$\sqrt{\beta} = \frac{1}{2\sqrt{2}} (-\lambda + \sqrt{\lambda^2 + 8\lambda}) \quad (6.5.3-17)$$

The expression of the ***velocity gain coefficient***  $\beta$  in terms of  $\lambda$  is

$$\boxed{\beta = \frac{1}{4} (\lambda^2 + 4\lambda - \lambda\sqrt{\lambda^2 + 8\lambda})} \quad (6.5.3-18)$$

Using (6.5.3-18) in (6.5.3-11) gives the ***position gain***  $\alpha$  in terms of  $\lambda$  as

$$\boxed{\alpha = -\frac{1}{8} (\lambda^2 + 8\lambda - (\lambda + 4)\sqrt{\lambda^2 + 8\lambda})} \quad (6.5.3-19)$$

The elements of the state estimation covariance matrix are, using (6.5.2-11),

$$p_{11} = (1 - g_1)m_{11} = g_1\sigma_w^2 \quad (6.5.3-20)$$

$$p_{12} = (1 - g_1)m_{12} = g_2\sigma_w^2 \quad (6.5.3-21)$$

$$p_{22} = \left( \frac{g_1}{T} + \frac{g_2}{2} \right) m_{12} - g_2 m_{12} = \left( \frac{g_1}{T} - \frac{g_2}{2} \right) m_{12} \quad (6.5.3-22)$$

The expressions of these (steady-state) error covariance matrix elements can be rewritten using (6.5.3-5) to (6.5.3-7) as

$$p_{11} = \alpha\sigma_w^2 \quad (6.5.3-23)$$

$$p_{12} = \frac{\beta}{T}\sigma_w^2 \quad (6.5.3-24)$$

$$p_{22} = \frac{\beta}{T^2} \frac{\alpha - \beta/2}{1 - \alpha} \sigma_w^2 \quad (6.5.3-25)$$

The (MS) **position estimation improvement**, or the **noise reduction factor**, with respect to a single observation is seen from (6.5.3-23) to be  $\alpha$  ( $0 \leq \alpha \leq 1$ ) — that is, the same as the *optimal* position gain of the filter.

The innovation variance is, in terms of the position prediction variance and the measurement noise variance,

$$s = m_{11} + \sigma_w^2 \quad (6.5.3-26)$$

Using (6.5.3-5) with  $g_1$  replaced by  $\alpha$  yields

$$s = \frac{\sigma_w^2}{1 - \alpha} \quad (6.5.3-27)$$

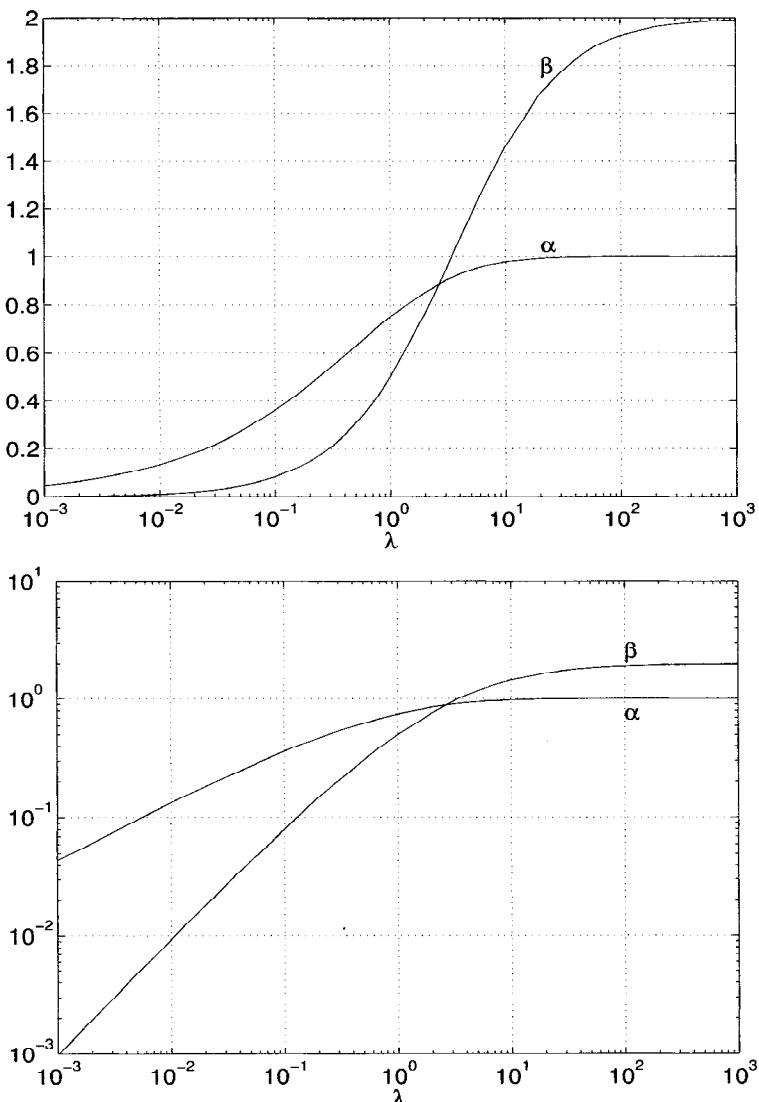
Figure 6.5.3-1 presents the **alpha-beta filter** gain coefficients  $\alpha$  and  $\beta$  as a function of the maneuvering index  $\lambda$  in semilog and log-log scales.

The **velocity estimation improvement** — compared to the differencing of two adjacent observations — is

$$\eta \triangleq \frac{p_{22}}{2\sigma_w^2/T^2} = \frac{\beta\alpha - \beta/2}{2} \frac{1}{1 - \alpha} \quad (6.5.3-28)$$

Note that this ignores the process noise — it is not meaningful for significant levels of the maneuvering index.

Figure 6.5.3-2 presents the velocity estimation improvement factor given above.



**Figure 6.5.3-1:** Steady-state filter gain coefficients for the piecewise constant white acceleration model.

## Simplified Expressions for Low Maneuvering Index

Note that for small  $\lambda$  (up to about 0.1) one has the following simplified expressions of the gains and the velocity estimation improvement factor

$$\alpha \approx \sqrt{2\lambda} \quad (6.5.3-29)$$

$$\beta \approx \lambda \quad (6.5.3-30)$$

$$\eta \approx \frac{1}{\sqrt{2}} \lambda^{1.5} \quad (6.5.3-31)$$

### Remarks

A high value of the process noise variance relative to the measurement noise variance — that is, a large maneuvering index  $\lambda$  — yields a high position gain  $\alpha$  and the filter will give large weight to the latest measurement and consequently little weight to the past data, resulting in less noise reduction.

A small  $\lambda$  yields a lower  $\alpha$  and more noise reduction. However, a small  $\alpha$  will *not* yield more noise reduction unless it has been *optimally determined* based on  $\lambda$  and *all the modeling assumptions hold*.

The two gains  $\alpha$  and  $\beta$  *cannot be chosen independently* — they are both determined by the maneuvering index  $\lambda$ .

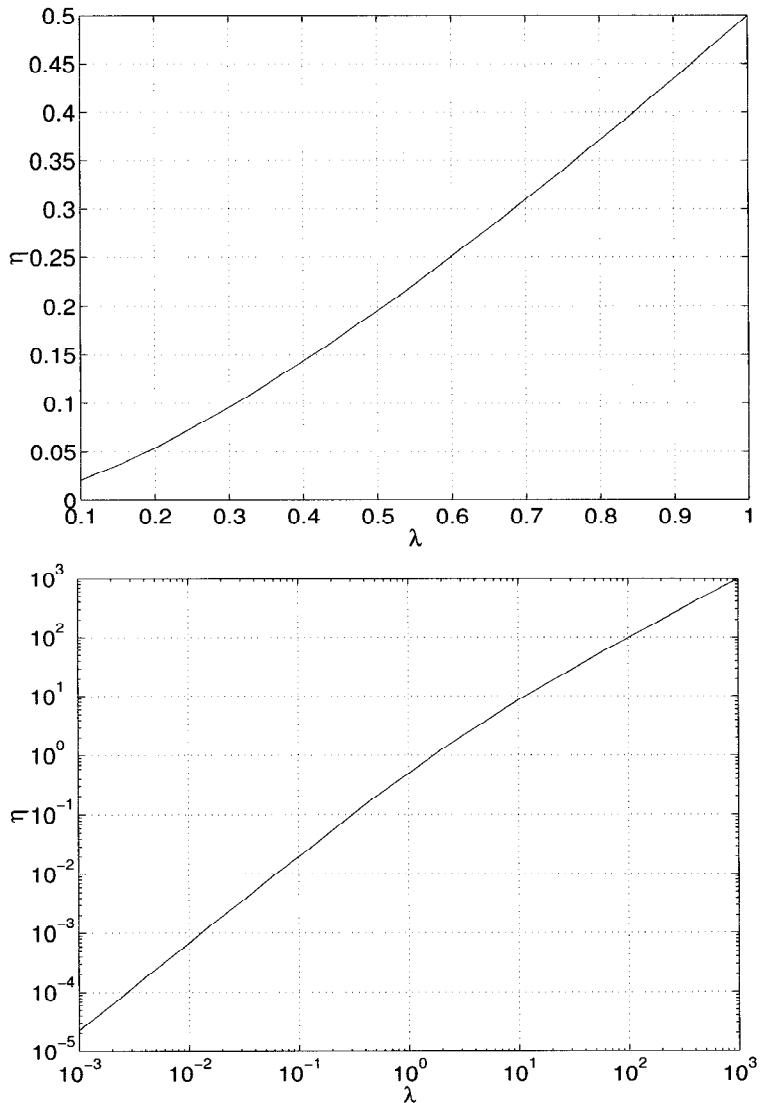
### Example

The example of Section 5.3, which dealt with what now is called a discrete white noise acceleration (DWNA) model, is reconsidered. The closed-form expressions for the steady-state gain and covariances developed above will be used and compared with the results of the covariance equation iterations that are plotted in Figs. 5.3.2-2 and 5.3.2-3.

The two cases of interest are those with nonzero process noise:  $q = 1$  and  $q = 9$  (the case with  $q = 0$  leads to zero variances and gain in steady state). The corresponding maneuvering indices are, using (6.5.3-14) with  $T = 1$ ,  $\sigma_v = \sqrt{q}$  and  $\sigma_w = 1$ ,  $\lambda = 1$  and  $\lambda = 3$ , respectively.

Table 6.5.3-1 shows the steady-state values of the gain coefficients as well as the position and velocity variances plotted in Figures 5.3.2-2 and 5.3.2-3, parts (b) and (c), respectively. The innovation variance (6.5.3-27) is also shown.

The updated variances, obtained from (6.5.3-23) and (6.5.3-25), and the predicted variances, obtained from (6.5.3-5) and (6.5.3-7), are seen to match the values plotted in the above figures.



**Figure 6.5.3-2:** Steady-state filter velocity estimation improvement over two-point differencing for the piecewise constant white acceleration model.

**Table 6.5.3-1:** Steady-state gains and variances for two maneuvering indices.

Maneuvering Index	Position Gain	Velocity Gain	Position Variance Updated	Position Variance Predicted	Velocity Variance Updated	Velocity Variance Predicted	Innovation Variance
$\lambda$	$\alpha$	$\beta$	$p_{11}$	$m_{11}$	$p_{22}$	$m_{22}$	$s$
1	0.75	0.50	0.75	3	1	2	4
3	0.90	0.94	0.90	9.15	4.12	13.12	10.15

### 6.5.4 The Alpha-Beta Filter for the Discretized CWNA Model

Next, the *alpha-beta filter* for the *discretized continuous-time white noise acceleration (CWNA) model* is derived using the process noise covariance matrix (6.2.2-12) instead of (6.3.2-4). Since almost all the equations stay the same as in Subsection 6.5.3, they will not be repeated and only those which are different will be indicated.

Equating the terms in the updated covariance expressions (6.5.2-11) and (6.5.2-12) with  $Q$  given by (6.2.2-12) yields

$$g_1 m_{11} = 2Tm_{12} - T^2 m_{22} + \frac{T^3}{3} \tilde{q} \quad (6.5.4-1)$$

$$g_1 m_{12} = Tm_{22} - \frac{T^2}{2} \tilde{q} \quad (6.5.4-2)$$

$$g_2 m_{12} = T\tilde{q} \quad (6.5.4-3)$$

To eliminate  $m_{22}$ , one has

$$m_{22} = \frac{g_1 m_{12}}{T} + \frac{T}{2} \tilde{q} = \left( \frac{g_1}{T} + \frac{g_2}{2} \right) m_{12} \quad (6.5.4-4)$$

The counterpart of (6.5.3-8) is

$$\frac{g_1^2}{1-g_1} \sigma_w^2 = 2T \frac{g_2}{1-g_1} \sigma_w^2 - T^2 \left( \frac{g_1}{T} + \frac{g_2}{2} \right) \frac{g_2}{1-g_1} \sigma_w^2 + \frac{T^2}{3} \frac{g_2^2}{1-g_1} \sigma_w^2 \quad (6.5.4-5)$$

which becomes

$$g_1^2 - 2Tg_2 + Tg_1g_2 + \frac{T^2}{6} g_2^2 = 0 \quad (6.5.4-6)$$

In terms of  $\alpha$  and  $\beta$ , the above can be written as

$$\alpha^2 - 2\beta + \alpha\beta + \frac{\beta^2}{6} = 0 \quad (6.5.4-7)$$

or

$$\alpha = \sqrt{2\beta + \frac{\beta^2}{12} - \frac{\beta}{2}} \quad (6.5.4-8)$$

The above equation differs from (6.5.3-11) by an extra term.

Equating  $m_{12}$  from (6.5.4-3) and (6.5.3-6) yields

$$m_{12} = \frac{T\tilde{q}}{\beta/T} = \frac{\beta/T}{1-\alpha}\sigma_w^2 \quad (6.5.4-9)$$

which results in

$$\frac{\beta^2}{1-\alpha} = \frac{T^3\tilde{q}}{\sigma_w^2} \triangleq \lambda_c^2 \quad (6.5.4-10)$$

where  $\lambda_c$  is the **maneuvering index** for this *discretized continuous-time system* and has a similar interpretation as  $\lambda$  in (6.5.3-14).

The equation for  $\beta$  becomes

$$\frac{\beta^2}{1-\alpha} = \frac{\beta^2}{1 - \sqrt{2\beta + \beta^2/12 + \beta/2}} = \lambda_c^2 \quad (6.5.4-11)$$

which can be solved explicitly as follows.

Let

$$u \triangleq \frac{\alpha^2}{\beta^2} \quad (6.5.4-12)$$

Using (6.5.4-12) in (6.5.4-8) yields

$$\beta = \frac{12}{6(u + \sqrt{u}) + 1} \quad (6.5.4-13)$$

and, thus,

$$\alpha = \beta\sqrt{u} = \frac{12\sqrt{u}}{6(u + \sqrt{u}) + 1} \quad (6.5.4-14)$$

Substituting (6.5.4-13) and (6.5.4-14) in (6.5.4-10) yields a quadratic equation for  $u$  with positive solution

$$u = \frac{1}{2} + \sqrt{\frac{1}{12} + \frac{4}{\lambda_c^2}} \quad (6.5.4-15)$$

which, when inserted in (6.5.4-13) and (6.5.4-14), provides the explicit solution for the gains of the alpha-beta filter for the discretized CWNA model.

Figure 6.5.4-1 presents the **alpha-beta filter** gain coefficients  $\alpha$  and  $\beta$  for the discretized CWNA model as a function of the maneuvering index  $\lambda_c$  in semilog and log-log scales.

The equations for the updated covariance terms (6.5.3-23) to (6.5.3-25) stay the same.

### Remark

This discretized continuous-time model is somewhat less common in use than the direct discrete-time piecewise constant acceleration model.

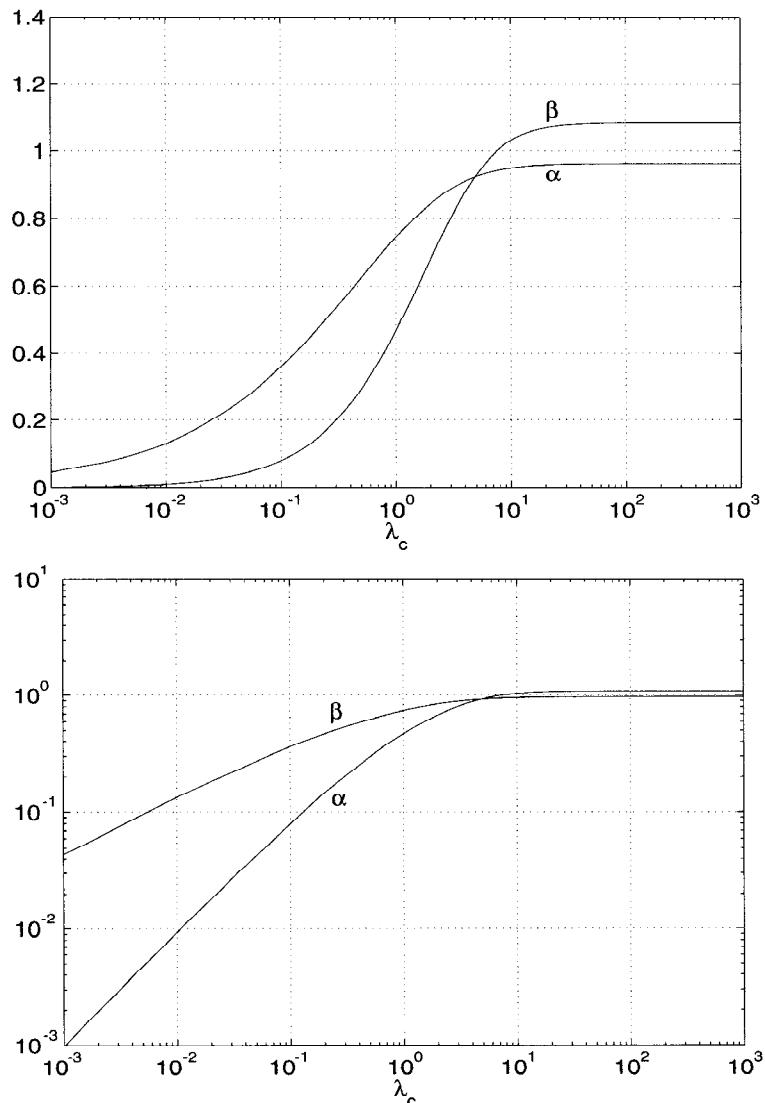


Figure 6.5.4-1: Steady-state filter gain coefficients for the discretized continuous-time white acceleration model.

### 6.5.5 The Alpha-Beta-Gamma Filter for the DWPA Model

The *piecewise constant Wiener process acceleration model* is the third-order system (6.3.3-1) with the *zero-mean white* process noise, assumed to be the *acceleration increment over a sampling period*, with variance  $\sigma_v^2$ .

The target maneuvering index is defined in the same manner as for the second-order model in (6.5.3-14), that is,

$$\lambda = \frac{\sigma_v T^2}{\sigma_w} \quad (6.5.5-1)$$

The steady-state gain for the resulting filter — the *alpha-beta-gamma filter* — is

$$\lim_{k \rightarrow \infty} W(k) \triangleq [g_1 \ g_2 \ g_3]' \triangleq \left[ \alpha \ \frac{\beta}{T} \ \frac{\gamma}{2T^2} \right]' \quad (6.5.5-2)$$

It can be shown that the three equations that yield the optimal steady-state filter *gain coefficients* are

$$\frac{\gamma^2}{4(1-\alpha)} = \lambda^2 \quad (6.5.5-3)$$

$$\beta = 2(2-\alpha) - 4\sqrt{1-\alpha} \quad (6.5.5-4)$$

or

$$\alpha = \sqrt{2\beta} - \frac{\beta}{2} \quad (6.5.5-5)$$

and

$$\gamma = \frac{\beta^2}{\alpha} \quad (6.5.5-6)$$

The relationship between  $\alpha$  and  $\beta$  in (6.5.5-4) is the same as (6.5.3-11).

The explicit solution for this system of three nonlinear equations that yields the three gain coefficients from (6.5.5-2) in terms of  $\lambda$  is given at the end of this subsection.

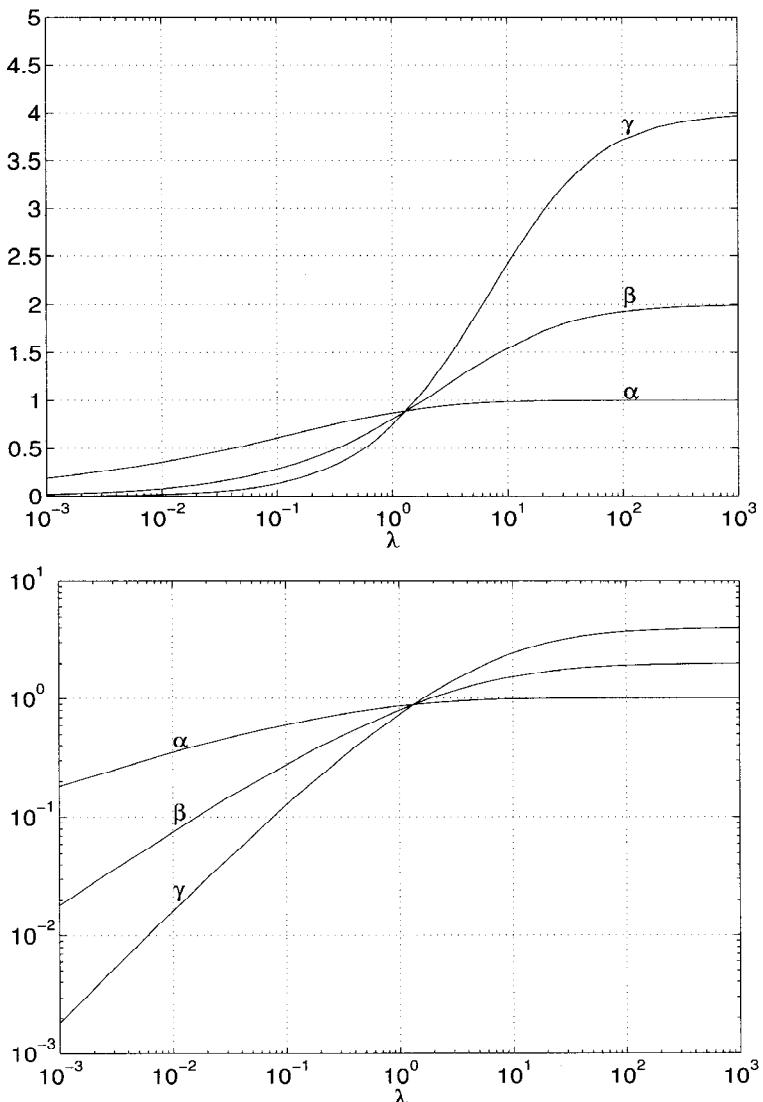
Figure 6.5.5-1 shows the gain coefficients for this filter as a function of the maneuvering index  $\lambda$  in semilog and log-log scale.

Similarly to the second-order system, it can be shown that the corresponding updated state covariance expressions (in steady state) are

$$p_{11} = \alpha \sigma_w^2 \quad p_{12} = \frac{\beta}{T} \sigma_w^2 \quad p_{13} = \frac{\gamma}{2T^2} \sigma_w^2 \quad (6.5.5-7)$$

$$p_{22} = \frac{8\alpha\beta + \gamma(\beta - 2\alpha - 4)}{8T^2(1-\alpha)} \sigma_w^2 \quad (6.5.5-8)$$

$$p_{23} = \frac{\beta(2\beta - \gamma)}{4T^3(1-\alpha)} \sigma_w^2 \quad p_{33} = \frac{\gamma(2\beta - \gamma)}{4T^4(1-\alpha)} \sigma_w^2 \quad (6.5.5-9)$$



**Figure 6.5.5-1:** Steady-state filter gain coefficients for the piecewise constant Wiener process acceleration model.

## The Solution for the Gain Coefficients

Substituting

$$\alpha = 1 - s^2 \quad (6.5.5-10)$$

in (6.5.5-4) yields

$$\beta = 2(1 - s)^2 \quad (6.5.5-11)$$

Rewriting (6.5.5-3) with (6.5.5-10) yields

$$\gamma = 2\lambda\sqrt{1 - \alpha} = 2\lambda s \quad (6.5.5-12)$$

Equations (6.5.5-10) to (6.5.5-12) provide the *explicit solution for the gain coefficients* in terms of the new variable  $s$  for which a cubic equation is obtained next.

Substituting (6.5.5-12) and (6.5.5-10) into the second equation of (6.5.5-6) leads to

$$2\lambda s = \frac{4(1 - s)^4}{1 - s^2} \quad (6.5.5-13)$$

which can be rewritten as

$$s^3 + bs^2 + cs - 1 = 0 \quad (6.5.5-14)$$

where

$$b \triangleq \frac{\lambda}{2} - 3 \quad c \triangleq \frac{\lambda}{2} + 3 \quad (6.5.5-15)$$

Substitute again

$$s = y - \frac{b}{3} \quad (6.5.5-16)$$

to obtain

$$y^3 + py + q = 0 \quad (6.5.5-17)$$

where

$$p \triangleq c - \frac{b^2}{3} \quad q \triangleq \frac{2b^3}{27} - \frac{bc}{3} - 1 \quad (6.5.5-18)$$

Finally, one more substitution,

$$y = z - \frac{p}{3z} \quad (6.5.5-19)$$

yields

$$z^6 + qz^3 - \frac{p^2}{27} = 0 \quad (6.5.5-20)$$

which has the solution

$$z^3 = \frac{-q \pm \sqrt{q^2 + 4p^2/27}}{2} \quad (6.5.5-21)$$

from which the negative sign should be chosen.

Using (6.5.5-16) and (6.5.5-19) yields

$$s = z - \frac{p}{3z} - \frac{b}{3} \quad (6.5.5-22)$$

which can be used directly in (6.5.5-10) to (6.5.5-12) to obtain the gain coefficients  $\alpha$ ,  $\beta$ , and  $\gamma$ .

### 6.5.6 A System Design Example for Sampling Rate Selection

The following *system design* problem is considered:

- An object moves in one dimension and undergoes acceleration up to  $1 \text{ m/s}^2$  and whose position is measured at intervals of  $T$  with an additive zero mean white noise with standard deviation  $\sigma_w = 10 \text{ m}$ .
- It is desired to determine the *sampling rate*, i.e., the *revisit rate*  $1/T$  such that the steady-state position estimation RMS error  $\sqrt{p_{11}} = 0.5 \text{ m}$ .

It will be assumed that the tracking of this moving object is based on a DWNA model, as discussed in Subsection 6.5.3, with process noise standard deviation  $\sigma_v = 1 \text{ m/s}^2$ .

Based on (6.5.3-23), the requirement on the position error translates into a noise reduction factor

$$\alpha = \frac{p_{11}}{\sigma_w^2} = \frac{0.25}{100} = \frac{1}{400} \quad (6.5.6-1)$$

Note that the required RMS reduction factor of  $1/20$  becomes  $1/400$  for the MSE.

Using (6.5.3-29) and the expression of the maneuvering index (6.5.3-14), one has

$$2 \frac{\sigma_v T^2}{\sigma_w} = \alpha^2 \quad (6.5.6-2)$$

which yields

$$T = \alpha \sqrt{\frac{\sigma_w}{2\sigma_v}} = \frac{\sqrt{10}}{400\sqrt{2}} \text{ s} \quad (6.5.6-3)$$

This corresponds to a sampling rate of nearly 200 Hz, i.e., extremely high. The reason for this is that the MS noise reduction factor of  $1/400$  is hard to achieve: Even for a static problem (i.e., an object at a fixed location) it would require 400 measurements of the object's fixed position to obtain this reduction factor.

In other words, one can improve the position accuracy over the sensor accuracy, but, typically, not by such a large factor. This illustrates the limitations faced when a certain performance has to be met.

If the desired accuracy is  $\sqrt{p_{11}} = 5 \text{ m}$ , then  $\alpha = 1/4$  and the resulting sampling interval is  $T = \sqrt{10}/(4\sqrt{2}) \approx 0.5 \text{ s}$ , which amounts to a more reasonable revisit rate of about 2 Hz.

### Remarks

A similar approach can be used if the system requirement is given for the velocity accuracy.

The same approach can be also used for a *trade-off* study between, for example, the sensor accuracy  $\sigma_w$  and  $T$ .

One has to keep in mind that the above relies on the DWNA model, which, like any model, is *only an approximation of the reality*. Alternatively, one can use another stochastic model like the discretized CWNA model (Subsection 6.5.4) or a deterministic model (e.g., as discussed in Section 1.1.4).

Note that, if the process noise assumption for the piecewise constant DWNA model (6.3.2-1) are valid for a given  $T$ , then, strictly speaking, they cannot be valid for any other  $T$ . On the other hand, while the discretized CWNA model (6.2.2-9) is valid for any  $T$ , its underlying continuous-time model (6.2.2-6) is an idealization. In other words, neither approach is perfect — this is why the designer has to have a *good understanding of the underlying modeling assumptions*.

### 6.5.7 Alpha-Beta and Alpha-Beta-Gamma Filters — Summary

For discrete-time kinematic models

- with zero-mean white process noise that models
  - the acceleration (second-order model) or
  - the acceleration increments (third-order model — Wiener process acceleration)
- with noisy position measurements,

one has explicit expressions of the *steady-state filter gain and the corresponding covariance*.

These filters, called alpha-beta and alpha-beta-gamma, respectively, are the simplest possible: They use fixed precomputed (steady-state) gains. Consequently, they are *not optimal* during the initial transient period or if the noises are nonstationary.

The gains of these filters depend *only* on the *target maneuvering index*.

The *target maneuvering index* is defined as the ratio between the standard deviations (RMS values) of the following two uncertainties:

- Motion uncertainty — the position displacement over one sampling period due to the process noise (multiplied by 2) and
- Observation uncertainty — the (position) measurement noise.

These filters are usually used independently for each coordinate; however, one can encounter instability under certain extreme circumstances due to the errors introduced by the decoupling [Rogers88].

Two classes of models were discussed:

- Discretized continuous-time models based on *continuous-time zero-mean white noise*,
- Direct discrete-time models based on *piecewise constant zero-mean white noise* — that is, a *zero-mean white sequence*.

These noises model the uncertainties of the motion — acceleration or acceleration increments. These two classes exhibit a different dependence on the sampling period of the effect of the noises on the motion.

None of these assumptions can model exactly target maneuvers, which are neither zero mean nor white — actually they are not even random, but the state models (which have to be Markov processes) require the specification of some randomness.

Nevertheless, they have been used extensively in real-time implementations of target tracking as well as for frequency estimation in speech recognition and in navigation systems. All the GPS (global positioning system) receivers use such filters to “smooth” the indicated position, speed and course (estimate them).

In particular, such fixed-gain filters have proven to be useful in implementations where their very modest computational and memory requirements were a major consideration.

One convenient application of these explicit results is to obtain quick (but possibly dirty) evaluations of achievable tracking performance — the quality of estimation, measured by the steady-state error variances. They can also be used for selection of system parameters, e.g., sampling interval.

These kinematic models can also be used as elements in a set of models describing different target behavior modes in the context of multiple model estimation algorithms, to be discussed in Chapter 11.

## 6.6 NOTES AND PROBLEMS

### 6.6.1 Bibliographical Notes

The kinematic (polynomial) models for filtering date back to [Sklansky57, Bencdict62]. They have been extensively discussed in the literature, for example, [Wishner70], and several papers presented steady-state filters for them. In [Friedland73] analytical expressions of the position and velocity estimation accuracy with position measurements were given. The coupling between range and range-rate (Doppler) measurements is discussed in [Fitzgerald74]. Gain curves as a function of the maneuvering index were presented in [Fitzgerald80]. Closed-form solutions for the continuous-time and discrete-time filter with exponentially autocorrelated acceleration were presented in [Fitzgerald81]. Tracking accuracies with position and velocity measurements were derived in [Castella81]. Analytical solutions for the steady-state filter gain and covariance with position and velocity measurements were given in [Ekstrand83].

The explicit derivations presented in Section 6.5 using the target maneuvering index are based on [Kalata84]. Its generalization to the coordinate-coupled case can be found in [Li97]. The idea of target maneuvering index has been used in [Friedland73] and can be traced back to [Sittler64]. The derivation of the explicit solution for the alpha-beta-gamma filter is based on [Gray93], while the explicit solution of the gains for the discretized CWNA model is due to J. Gray.

Track initiation and simple approximations of the gains during the transient for kinematic models have been discussed in [Kalata84].

Frequency domain analysis of alpha-beta filters and the steady-state bias resulting from constant accelerations are discussed in [Farina85]. The equivalent bandwidth of polynomial filters has been presented in [Ng83]. The response of alpha-beta-gamma filters to step inputs can be found in [Navarro77].

A discussion of the possible unbounded errors in decoupled alpha-beta filters is presented in [Rogers88].

## 6.6.2 Problems

- 6-1 Simulated kinematic trajectory behavior.** A target is simulated as having a nearly constant velocity motion with white noise acceleration, constant over the sampling period  $T$ , as in (6.3.2-1). The noise is  $v(k) \sim \mathcal{N}(0, \sigma_v^2)$ .
1. Find the prior pdf of the velocity at time  $k$ .
  2. What is the range of the velocity  $k$  sampling periods after the initial time?
  3. Assume that the initial velocity is 10, and that the process noise variance and the sampling time are both unity. If after  $k = 25$  samples the velocity became zero, is this a sign that the random number generator is biased?
  4. What conclusion can be drawn from the above about the behavior of the velocity for the third-order kinematic model (6.3.3-1) with process noise representing acceleration increments?
- 6-2 Process noise rescaling** (for second-order direct discrete time kinematic model when sampling period is changed). A target is simulated according to the second-order kinematic model (6.3.2-1) with a sampling period  $T_1$  with process noise  $v(k)$ , which represents the constant acceleration over a sampling period, with variance  $\sigma_v^2(T_1)$ . Subsequently, the sampling period is changed to  $T_2$  and we want to preserve the statistical properties of the motion. What should be done?
- 6-3 Process noise rescaling** (for third-order direct discrete time kinematic model when sampling period is changed). A target is simulated according to the third-order kinematic model (6.3.3-1) with a sampling period  $T_1$  with process noise  $v(k)$ , which represents the acceleration increment over a sampling period, with variance  $\sigma_v^2(T_1)$ . Subsequently, the sampling period is changed to  $T_2$  and we want to preserve the statistical properties of the motion. What should be done?
- 6-4 Simulated kinematic trajectory variability when sampling period is changed.** A trajectory is generated as in problem 6-3 with  $T_1 = 1$  and  $\sigma_v^2(T_1) = 0.1$  for  $N_1 = 20$  periods. Then, using the same random number generator seed, the same trajectory is generated with a higher sampling rate,  $T_2 = 0.5$  for  $N_2 = 40$  samples — that is, the same total time.
1. Can one expect the trajectories to be identical?

2. Find the range of the difference between the two trajectories' accelerations at the common final time  $t_F$ .

#### 6-5 Alpha-beta filter design and evaluation.

1. Design an  $\alpha\text{-}\beta$  filter for a data rate of 40 Hz, with maximum acceleration  $|a_M| = 16g$  ( $g \approx 10 \text{ m/s}^2$ ), and measurement noise with  $\sigma_w = 10 \text{ m}$ .
2. Calculate the (steady-state) error covariance matrix elements and the position and velocity RMS errors.
3. Calculate the RMS position prediction error for a prediction time of  $t = 2 \text{ s}$  under the assumptions of the filter.
4. Assume that the target has a constant acceleration of  $16g$  during this prediction time. Calculate the position prediction error due to this and compare it with the result from item 3.

#### 6-6 Existence of steady-state filter for a kinematic model.

Consider the system

$$\begin{aligned} x(k+1) &= Fx(k) + \Gamma v(k) \\ z(k) &= Hx(k) + w(k) \end{aligned}$$

with

$$F = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad \Gamma = \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \quad H = [1 \ 0]$$

and the (scalar) process and measurement noise sequences are zero mean white with variances  $q$  and  $r$ , respectively.

1. State the condition for stability (existence of steady state) of the Kalman filter for this system in terms of  $F$  and  $H$ . Prove that this holds for the above system.
2. State the additional condition that the steady-state filter covariance is positive definite and unique. Prove that this also holds.

#### 6-7 Alpha filter.

Consider the scalar system

$$\begin{aligned} x(k+1) &= x(k) + \frac{T^2}{2}v(k) \\ z(k) &= x(k) + w(k) \end{aligned}$$

with the two noise sequences mutually uncorrelated, zero mean, white, and with variances  $\sigma_v^2$  and  $\sigma_w^2$ , respectively. Let

$$\lambda \triangleq \frac{T^2\sigma_v}{\sigma_w}$$

Find

1. The steady-state Kalman filter gain  $\alpha$  in terms of  $\lambda$ .
2. The noise reduction factor

$$\frac{P_{xx}}{\sigma_w^2}$$

in terms of  $\alpha$ , where  $P_{xx}$  is the steady-state variance of the estimate of  $x$  from the KF.

**6-8 Alpha filter and suboptimality.** Consider the scalar system

$$x(k+1) = x(k) + v(k)$$

with  $v(k)$  zero-mean, white with variance  $q = 1$  and the observations

$$z(k) = x(k) + w(k)$$

with  $w(k)$  zero-mean, white with variance  $r = 1$ , with the two sequences and the initial error mutually uncorrelated.

1. Find the optimal steady-state Kalman filter gain.
2. Find the MS and RMS noise reduction factors.
3. Assume the filter has the arbitrary gain  $W = 0.1$ . Find the steady state MS estimation error. How much is now the “noise reduction factor”?
4. Find the range of values of  $W$  for which the filter is stable.
5. What happens if  $W = 2$ ?

**6-9 Bias and total MS error in the alpha filter.** Consider the estimator (alpha filter)

$$\hat{x}(k+1) = \hat{x}(k) + \alpha[z(k+1) - \hat{x}(k)]$$

with  $0 < \alpha < 1$ , for the scalar system

$$x(k+1) = x(k) + v$$

where  $v$  is a fixed bias, with measurements

$$z(k) = x(k) + w(k)$$

where  $w(k)$  is a zero-mean white noise with known variance  $r$ .

1. Assume  $v = 0$ . Find the steady-state MSE

$$P = E[\tilde{x}(k)^2]$$

(due to the measurement noise only).

2. Assume  $v \neq 0$ . Find the steady-state bias

$$b = E[\tilde{x}(k)]$$

due to  $v$ .

3. Find the total MSE (due to both  $v$  and the measurement noise).

**6-10 Filter with position/velocity measurement.** Consider a DWNA model (with piecewise constant accelerations) for one coordinate with the measurement being a linear combination of the position  $\xi$  and velocity  $\dot{\xi}$ , namely,

$$z(k) = [1 \ a] \begin{bmatrix} \xi \\ \dot{\xi} \end{bmatrix} + w(k)$$

Assume one has the state covariance  $P(1|0) = I$ . Furthermore, let the sampling interval be  $T = 1$ , the process noise variance  $\sigma_v^2 = 0$  and the measurement noise variance  $\sigma_w^2 = 1$ .

1. Find  $P(2|1)$ .
2. If the choices for  $a$  are  $a > 0$  or  $a, 0$ , which one would you pick and why?
3. Find the optimal value of  $a$  for minimum position variance in  $P(2|1)$ .
4. Set up the equations for the steady state filter gains.
5. Solve them.
6. Find the optimal value of  $a$  for minimum position variance component in  $P(k|k)$  in steady state.

**6-11 Bandwidth of an alpha filter.** Consider the alpha filter for a scalar system with state  $x$  and observation  $y$

$$\dot{\hat{x}}(k+1) = a \dot{\hat{x}}(k) + \alpha [\dot{y}(k+1) - a \dot{\hat{x}}(k)]$$

1. Find the transfer function

$$H(z) = \frac{\hat{X}(z)}{\hat{Y}(z)}$$

2. With the sampling interval  $T$  and

$$z = e^{j\omega T} = e^{j2\pi fT} = e^{j2\pi f/f_0}$$

find the bandwidth of the filter, i.e., the frequency  $f$  (with  $|f| < f_0/2$ ) at which

$$|H(f)/H(f=0)|^2 = 1/2$$

Evaluate for  $a = 1$ ,  $\alpha = 0.5$ . You can assume  $\cos^{-1} 0.75 \approx 45^\circ$ . Express the resulting  $f_{BW}$  in terms of  $f_0 = 1/T$ .

**6-12 Bias and total MS error in the alpha-beta filter.** The state  $x$  of a target, consisting of range  $r$  and range rate  $\dot{r}$ , is observed with range measurements and is estimated with an  $\alpha\text{-}\beta$  filter at intervals  $T$ .

The filter has been designed for zero-mean white noises  $v(k)$ ,  $w(k)$ . However, there is a constant input (bias) target acceleration  $u$ . The goal is to find the total MSE in the estimates.

The dynamic equation is

$$x(k) = Fx(k-1) + G[u + v(k-1)]$$

where

$$F = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad G = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}$$

The measurement is

$$z(k) = [1 \ 0]x(k) + w(k) = Hx(k) + w(k)$$

The filter gain is

$$W = \begin{bmatrix} \alpha \\ \beta \\ T \end{bmatrix}$$

1. Show that the mean of the estimation error  $\bar{x}$  obeys a dynamic equation

$$\bar{x}(k) = \tilde{F}\bar{x}(k-1) + \tilde{G}u$$

Find  $\tilde{F}$  and  $\tilde{G}$ .

2. Find the expressions of the components of the steady-state mean error in terms of  $T$ ,  $\alpha$ ,  $\beta$  and  $u$ .
3. Evaluate the above for  $T = 0.25$  s,  $\alpha = 0.75$ ,  $\beta = 0.5$  and  $u = 5$  m/s<sup>2</sup>.
4. Find the total estimation MSE (in s.s.) assuming  $\sigma_w = 2$  m.

- 6-13 Position and velocity estimation for known acceleration objects.** Consider an object that moves in one dimension with an exactly known acceleration. Show that if a large number  $m$  of position measurements are made on this object, with errors that are zero-mean, white and with variance  $\sigma^2$ , at intervals  $T$  (covering a total time span  $T_T \approx mT$ ), then the RMS accuracies of its position and velocity estimates at the end of the interval are given by  $2\sigma/\sqrt{m}$  and  $2\sqrt{3}\sigma/(\sqrt{m}T_T)$ , respectively.

*Hint:* Recast the problem, with appropriate justification, so you can use the results from Subsection 6.4.2.



# Chapter 7

## COMPUTATIONAL ASPECTS OF ESTIMATION

### 7.1 INTRODUCTION

#### 7.1.1 Implementation of Linear Estimation

This chapter describes briefly some numerical techniques for the efficient *implementation of the linear estimation techniques* presented earlier.

The techniques to be discussed are for discrete-time linear systems with state equation

$$x(k+1) = F(k)x(k) + \Gamma(k)v(k) \quad (7.1.1-1)$$

and measurement equation

$$z(k) = H(k)x(k) + w(k) \quad (7.1.1-2)$$

where  $v(k)$  and  $w(k)$  are the process and measurement noises, assumed to be zero mean, white, mutually uncorrelated, and with covariances  $Q(k)$  and  $R(k)$ , respectively.

The two properties of a covariance matrix

- symmetry and
- positive definiteness

can be lost due to *round-off errors* in the course of calculating its propagation equations — the covariance prediction and the covariance update.

The covariance propagation equations are discussed next, and their propensity for causing *loss of symmetry* and/or *loss of positive definiteness* is examined.

The **covariance prediction equation**

$$P(k+1|k) = F(k)P(k|k)F(k)' + \Gamma(k)Q(k)\Gamma(k)' \quad (7.1.1-3)$$

can affect only the symmetry of the resulting matrix. A suitable implementation of the products of three matrices will avoid this problem.

More significant numerical problems arise in the **covariance update equation**, which can be written in the following algebraically equivalent forms:

$$P(k+1|k+1) = [I - W(k+1)H(k+1)]P(k+1|k) \quad (7.1.1-4)$$

$$P(k+1|k+1) = P(k+1|k) - W(k+1)S(k+1)W(k+1)' \quad (7.1.1-5)$$

$$\begin{aligned} P(k+1|k+1) &= [I - W(k+1)H(k+1)]P(k+1|k) \\ &\cdot [I - W(k+1)H(k+1)]' + W(k+1)R(k+1)W(k+1)' \end{aligned} \quad (7.1.1-6)$$

Equation (7.1.1-4) is very sensitive to round-off errors and is bound to lead to loss of symmetry as well as positive definiteness — it is best to avoid it.

Equation (7.1.1-5) will avoid loss of symmetry with a suitable implementation of the last term, which is a product of three matrices (as in the covariance prediction), but it can still lead to loss of positive definiteness due to numerical errors in the subtraction.

Equation (7.1.1-6), the **Joseph form covariance update**, while computationally more expensive than (7.1.1-5), is less sensitive to round-off errors. With the proper implementation of the products of three matrices, it will preserve symmetry. Furthermore, since the only place it has a subtraction is in the term  $I - WH$ , which appears “squared,” this form of the covariance update has the property of preserving the positive definiteness of the resulting updated covariance.

## 7.1.2 Outline

The purposes of the techniques to be discussed in this chapter are:

1. Reduction of the computational requirements.
2. Improvement of the numerical accuracy — preservation of the symmetry and the positive definiteness of the state covariance matrix, as well as reduction of its **condition number**, defined as the logarithm of the ratio of its largest to its smallest eigenvalue in (1.3.6-13).

Section 7.2 presents the **information filter**, which carries out the recursive computation of the inverse of the covariance matrix. This is an alternative to (7.1.1-5) and is less demanding computationally for systems with dimension of the measurement vector larger than that of the state. This technique also has the advantage that it allows the *start-up of the estimation without an initial estimate*. Specifically, this amounts to having a state estimator with a **noninformative (diffuse) prior**, similar to the one discussed in Section 2.3.4 for the parameter estimation case.

A technique which carries out the state update with one measurement component at a time — the *sequential updating* technique — rather than the entire measurement vector at once, is described in Section 7.3. This implementation of the Kalman filter is less demanding computationally than the standard one.

The technique of *square-root filtering*, which consists of *sequential updating* and *factorization of the covariance matrix*, is described in Section 7.4. This approach avoids the numerical problems that can lead, due to round-off errors, to the loss of symmetry and positive definiteness of the state covariance matrix. It is also less expensive computationally than the best standard implementation (7.1.1-6) while it has double the accuracy.

### 7.1.3 Computational Aspects — Summary of Objectives

Present techniques for implementation of the Kalman filter that are

- more economical computationally and/or
- more robust numerically.

Typical numerical problems with the state covariance matrix:

- Loss of symmetry.
- Loss of positive definiteness.

The algorithms to be discussed:

- Information filter — calculates recursively the inverse covariance matrix and it also allows to start the state estimator without an initial estimate, i.e., with a noninformative prior.
- Sequential processing of measurements in updating the state estimate.
- Square-root filter: a combination of sequential updating with covariance factorization that achieves double precision.

## 7.2 THE INFORMATION FILTER

### 7.2.1 Recursions for the Information Matrices

The standard version of the Kalman filter calculates the gain in conjunction with a recursive computation of the state covariance. Starting from  $P(k-1|k-1)$ , the prediction covariance is obtained as

$$P(k|k-1) = F(k-1)P(k-1|k-1)F(k-1)' + \Gamma(k-1)Q(k-1)\Gamma(k-1)' \quad (7.2.1-1)$$

The gain is

$$W(k) = P(k|k-1)H(k)'[H(k)P(k|k-1)H(k)' + R(k)]^{-1} \quad (7.2.1-2)$$

and the updated state covariance  $P(k|k)$  is calculated from

$$\begin{aligned} P(k|k) &= P(k|k-1) - P(k|k-1)H(k)'[H(k)P(k|k-1)H(k)' \\ &\quad + R(k)]^{-1}H(k)P(k|k-1) \end{aligned} \quad (7.2.1-3)$$

which completes one cycle of computations.

The **information filter** calculates recursively the *inverses of the covariance matrices*, both for the prediction and the update. The term “information” is used in the sense of the Cramer-Rao lower bound, where the (**Fisher**) **information matrix** is the *inverse of the covariance matrix*.

A recursion from  $P(k-1|k-1)^{-1}$  to  $P(k|k-1)^{-1}$  to  $P(k|k)^{-1}$  and the expression of the filter gain in terms of (one of) the above inverses are presented next.

### The update

The update for the information matrix follows immediately from (7.2.1-3) and the matrix inversion lemma (see Subsection 1.3.3) as

$$P(k|k)^{-1} = P(k|k-1)^{-1} + H(k)'R(k)^{-1}H(k) \quad (7.2.1-4)$$

### The prediction

Denote the information matrix corresponding to the state prediction *without process noise* as

$$A(k-1)^{-1} \triangleq F(k-1)P(k-1|k-1)F(k-1)' \quad (7.2.1-5)$$

or, since  $F$ , as a transition matrix, is invertible,

$$A(k-1) = [F(k-1)^{-1}]'P(k-1|k-1)^{-1}F(k-1)^{-1} \quad (7.2.1-6)$$

The prediction information matrix is

$$\begin{aligned} P(k|k-1)^{-1} &= [F(k-1)P(k-1|k-1)F(k-1)' + \Gamma(k-1) \\ &\quad \cdot Q(k-1)\Gamma(k-1)']^{-1} \\ &= [A(k-1)^{-1} + \Gamma(k-1)Q(k-1)\Gamma(k-1)']^{-1} \end{aligned} \quad (7.2.1-7)$$

and it can be rewritten, again with the matrix inversion lemma, as

$$\begin{aligned} P(k|k-1)^{-1} &= A(k-1) - A(k-1)\Gamma(k-1) \\ &\quad \cdot [\Gamma(k-1)'A(k-1)\Gamma(k-1) + Q(k-1)^{-1}]^{-1}\Gamma(k-1)'A(k-1) \end{aligned} \quad (7.2.1-8)$$

The expression of the gain (7.2.1-2) is rewritten as

$$\begin{aligned}
 W(k) &= P(k|k-1)H(k)' \left\{ [H(k)P(k|k-1)H(k)' + R(k)]^{-1} \right. \\
 &\quad \left. + R(k)^{-1} - R(k)^{-1} \right\} \\
 &= P(k|k-1)H(k)'R(k)^{-1} \\
 &\quad + P(k|k-1)H(k)'[H(k)P(k|k-1)H(k)' + R(k)]^{-1} \\
 &\quad \cdot \left\{ I - [H(k)P(k|k-1)H(k)' + R(k)]R(k)^{-1} \right\} \\
 &= \left\{ P(k|k-1) - P(k|k-1)H(k)'[H(k)P(k|k-1)H(k)' + R(k)]^{-1} \right. \\
 &\quad \left. \cdot H(k)P(k|k-1) \right\} H(k)'R(k)^{-1}
 \end{aligned} \tag{7.2.1-9}$$

With the matrix inversion lemma, (7.2.1-9) becomes

$$W(k) = [P(k|k-1)^{-1} + H(k)'R(k)^{-1}H(k)]^{-1}H(k)'R(k)^{-1} \tag{7.2.1-10}$$

which is the sought-after expression of the gain. It can be easily shown that the above is equivalent to the alternate form of the gain:

$$W(k) = P(k|k)H(k)'R(k)^{-1} \tag{7.2.1-11}$$

### Duality Between the Covariance and Information Equations

Note the **duality**<sup>1</sup> between the covariance and the information propagation equations:

$$\begin{array}{ccc}
 \text{Covariance prediction (7.2.1-1)} & \longleftrightarrow & \text{Information update (7.2.1-4)} \\
 \text{Covariance update (7.2.1-3)} & \longleftrightarrow & \text{Information prediction (7.2.1-8)}
 \end{array}$$

Carrying out in (7.2.1-1), with the time index increased by one, the following replacements

$$P(k+1|k) \rightarrow P(k+1|k+1)^{-1} \tag{7.2.1-12}$$

$$A(k) \rightarrow P(k+1|k) \tag{7.2.1-13}$$

$$\Gamma(k) \rightarrow H(k)' \tag{7.2.1-14}$$

$$Q(k) \rightarrow R(k)^{-1} \tag{7.2.1-15}$$

yield (7.2.1-4).

Similarly, carrying out in (7.2.1-3), again with the time index increased by one, the replacements

$$P(k+1|k)^{-1} \leftarrow P(k+1|k+1) \tag{7.2.1-16}$$

---

<sup>1</sup>The concept of duality can be illustrated by what the frog said in a restaurant when he got his soup: "Waiter! There is *no* fly in my soup!"

$$A(k) \leftarrow P(k+1|k) \quad (7.2.1-17)$$

$$\Gamma(k) \leftarrow H(k)' \quad (7.2.1-18)$$

$$Q(k)^{-1} \leftarrow R(k) \quad (7.2.1-19)$$

yield (7.2.1-8).

Since (7.2.1-16) to (7.2.1-19) are exactly the inverses of (7.2.1-12) to (7.2.1-15), this proves the duality.

## 7.2.2 Overview of the Information Filter Algorithm

The sequence of calculations of the information matrices and filter gain for one cycle of the information filter are presented in Fig. 7.2.2-1.

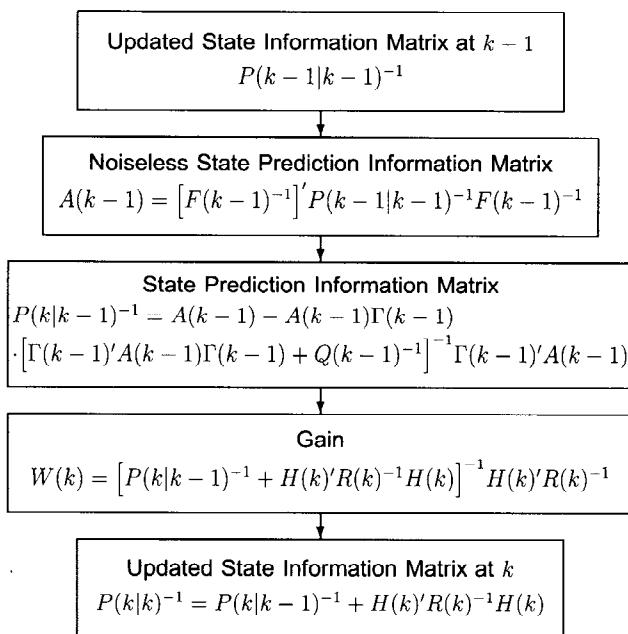


Figure 7.2.2-1: The information matrix and filter gain calculations in the information filter.

The state estimate  $\hat{x}$  can be computed in the same manner as in the conventional Kalman filter. Namely, Fig. 7.2.2-1 replaces the right-hand column of computations from Fig. 5.2.4-1. There is an alternative, however, that is discussed in the next subsection.

This implementation of the Kalman filter is advantageous when the dimension  $n_z$  of the measurement vector is larger than the dimension  $n_x$  of the state. Note that the inversions in the above sequence of calculations are for  $n_x \times n_x$  matrices (and  $Q$ , which is  $n_v \times n_v$ , with  $n_v \leq n_x$ ), while the conventional algorithm requires the inversion of the innovation covariance, which is  $n_z \times n_z$ . The inverse of the measurement noise covariance  $R(k)$  is usually simple to obtain because it is, in many cases, diagonal.

### 7.2.3 Recursion for the Information Filter State

For the situations where there is no initial estimate, one can start the estimation with the initial information matrix as zero, namely,

$$P(0|0)^{-1} = 0 \quad (7.2.3-1)$$

This amounts to a **noninformative (diffuse) prior** because of the infinite uncertainty (infinite variance) associated with it.

In this case one can obtain recursions from

$$\hat{y}(k-1|k-1) \triangleq P(k-1|k-1)^{-1}\hat{x}(k-1|k-1) \quad (7.2.3-2)$$

to

$$\hat{y}(k|k-1) \triangleq P(k|k-1)^{-1}\hat{x}(k|k-1) \quad (7.2.3-3)$$

and then to

$$\hat{y}(k|k) \triangleq P(k|k)^{-1}\hat{x}(k|k) \quad (7.2.3-4)$$

Note that, in view of (7.2.3-1), the initial value of  $\hat{y}$ , to be called the **information filter state**, is  $\hat{y}(0|0) = 0$ , regardless of  $\hat{x}(0|0)$  (as long as it is finite), i.e., *no initial estimate  $\hat{x}(0|0)$  of the system's state is needed*.

### The prediction

Using (7.2.3-3), (7.2.1-1) and the matrix inversion lemma, one has

$$\begin{aligned} \hat{y}(k|k-1) &= P(k|k-1)^{-1}\hat{x}(k|k-1) \\ &= [F(k-1)P(k-1|k-1)F(k-1)']^{-1} \\ &\quad + \Gamma(k-1)Q(k-1)\Gamma(k-1)'^{-1}F(k-1)\hat{x}(k-1|k-1) \\ &= [F(k-1)']^{-1}[P(k-1|k-1) + F(k-1)^{-1}\Gamma(k-1)Q(k-1) \\ &\quad \cdot \Gamma(k-1)'(F(k-1)')^{-1}]^{-1}F(k-1)^{-1}F(k-1)\hat{x}(k-1|k-1) \\ &= [F(k-1)']^{-1}\{P(k-1|k-1)^{-1} - P(k-1|k-1)^{-1} \\ &\quad \cdot F(k-1)^{-1}\Gamma(k-1)[\Gamma(k-1)'(F(k-1)')^{-1}P(k-1|k-1)^{-1} \\ &\quad \cdot F(k-1)^{-1}\Gamma(k-1) + Q(k-1)^{-1}]^{-1} \\ &\quad \cdot \Gamma(k-1)'(F(k-1)')^{-1}P(k-1|k-1)^{-1}\}\hat{x}(k-1|k-1) \\ &= [F(k-1)']^{-1}\{I - P(k-1|k-1)^{-1}F(k-1)^{-1}\Gamma(k-1) \\ &\quad \cdot [\Gamma(k-1)'(F(k-1)')^{-1}P(k-1|k-1)^{-1}F(k-1)^{-1}\Gamma(k-1) \\ &\quad + Q(k-1)^{-1}]^{-1}\Gamma(k-1)'(F(k-1)')^{-1}\hat{y}(k-1|k-1)\} \quad (7.2.3-5) \end{aligned}$$

Using the notation (7.2.1-5), the above can be rewritten as

$$\boxed{\hat{y}(k|k-1) = (F(k-1)')^{-1} \{ I - P(k-1|k-1)^{-1} F(k-1)^{-1} \Gamma(k-1) \\ \cdot [\Gamma(k-1)' A(k-1) \Gamma(k-1) + Q(k-1)^{-1}]^{-1} \\ \cdot \Gamma(k-1)' (F(k-1)')^{-1} \hat{y}(k-1|k-1) \}} \quad (7.2.3-6)$$

### The update

Using (7.2.3-4) and (7.2.1-4), one can write

$$\begin{aligned} \hat{y}(k|k) &= P(k|k)^{-1} \hat{x}(k|k) \\ &= [P(k|k-1)^{-1} + H(k)' R(k)^{-1} H(k)] \hat{x}(k|k-1) \\ &\quad + P(k|k)^{-1} P(k|k) H(k)' R(k)^{-1} [z(k) - H(k) \hat{x}(k|k-1)] \\ &= P(k|k-1)^{-1} \hat{x}(k|k-1) + H(k)' R(k)^{-1} z(k) \end{aligned} \quad (7.2.3-7)$$

which can be rewritten as

$$\boxed{\hat{y}(k|k) = \hat{y}(k|k-1) + H(k)' R(k)^{-1} z(k)} \quad (7.2.3-8)$$

Once  $P(k|k)^{-1}$  becomes invertible, i.e., all the uncertainties about the state are finite, one can recover the state estimate from the information filter state as follows:

$$\hat{x}(k|k) = P(k|k) \hat{y}(k|k) \quad (7.2.3-9)$$

## 7.3 SEQUENTIAL PROCESSING OF MEASUREMENTS

### 7.3.1 Block vs. Sequential Processing

The standard implementation of the Kalman filter for a vector measurement is by carrying out the state update simultaneously with the entire measurement vector from a given time, i.e., **block processing**.

If the measurement noise vector components  $w_i(k)$ ,  $i = 1, \dots, n_z$ , are uncorrelated, that is,

$$R(k) = E[w(k)w(k)'] = \text{diag}[r_1(k), \dots, r_{n_z}(k)] \quad (7.3.1-1)$$

then one can carry out the update of the state with *one component of the measurement at a time*, that is, **sequential processing** or **scalar updates**.

If the matrix  $R$  is not diagonal, one can apply a linear transformation on the measurement to diagonalize its covariance matrix. This can be done using a  $LDL'$  factorization, to be discussed later in Subsection 7.4.2.

The measurement

$$\begin{aligned} z(k) &= \begin{bmatrix} z_1(k) \\ \vdots \\ z_{n_z}(k) \end{bmatrix} \\ &= H(k)x(k) + w(k) = \begin{bmatrix} h_1(k)'x(k) + w_1(k) \\ \vdots \\ h_{n_z}(k)'x(k) + w_{n_z}(k) \end{bmatrix} \end{aligned} \quad (7.3.1-2)$$

will be considered as a *sequence of scalar measurements*  $z_i(k)$ ,  $i = 1, \dots, n_z$ .

The uncorrelatedness of the corresponding measurement noises allows the use of the Kalman filter update sequentially for each “scalar measurement” because it implies whiteness for the scalar measurement noise “sequence”

$$w_1(k), \dots, w_{n_z}(k), w_1(k+1), \dots, w_{n_z}(k+1), \dots \quad (7.3.1-3)$$

A natural application of this is in a **multisensor** situation where the measurement noises are independent across sensors.

### 7.3.2 The Sequential Processing Algorithm

The sequence of *scalar updates* is described below.

Starting from the predicted state to  $k$  from  $k-1$ , denoted as

$$\hat{x}(k|k, 0) \triangleq \hat{x}(k|k-1) \quad (7.3.2-1)$$

with associated covariance

$$P(k|k, 0) \triangleq P(k|k-1) \quad (7.3.2-2)$$

the following sequence of calculations is carried out for  $i = 1, \dots, n_z$ :

The scalar innovation corresponding to  $z_i(k)$  has variance

$$s(k, i) = h_i(k)'P(k|k, i-1)h_i(k) + r_i(k) \quad (7.3.2-3)$$

The corresponding gain is

$$W(k, i) = \frac{P(k|k, i-1)h_i(k)}{s(k, i)} \quad (7.3.2-4)$$

and the updated state is

$$\hat{x}(k|k, i) = \hat{x}(k|k, i-1) + W(k, i)[z_i(k) - h_i(k)'\hat{x}(k|k, i-1)] \quad (7.3.2-5)$$

with covariance

$$P(k|k, i) = P(k|k, i-1) - W(k, i)h_i(k)'P(k|k, i-1) \quad (7.3.2-6)$$

which can be rewritten (and should be implemented) as

$$P(k|k, i) = P(k|k, i-1) - \frac{P(k|k, i-1)h_i(k)h_i(k)'P(k|k, i-1)}{h_i(k)'P(k|k, i-1)h_i(k) + r_i(k)} \quad (7.3.2-7)$$

Note that, in the  $i$ th update,  $P(k|k, i-1)$  plays the role of a prediction covariance. Finally,

$$\hat{x}(k|k) \triangleq \hat{x}(k|k, n_z) \quad (7.3.2-8)$$

$$P(k|k) \triangleq P(k|k, n_z) \quad (7.3.2-9)$$

Note that this approach eliminates the need for the information filter since only scalar inversions are performed here.

If the measurement noise covariance matrix (7.3.1-1) is not diagonal, a transformation that diagonalizes its covariance matrix can be used. An efficient technique to accomplish this is via the  $LDL'$  factorization, to be discussed in detail in Section 7.4. This factorization yields

$$R = LDL' \quad (7.3.2-10)$$

where  $L$  is a *lower triangular matrix* and  $D_R$  is a *diagonal matrix with positive entries*. Instead of  $z$ , one can use the transformed measurement

$$\bar{z} = L^{-1}z \quad (7.3.2-11)$$

and instead of  $H$  one has

$$\bar{H} = L^{-1}H \quad (7.3.2-12)$$

This then allows the use of the scalar update procedure described above.

## A Note on Numerical Accuracy

The covariance update (7.3.2-7) has a subtraction and, therefore, is susceptible to numerical errors, which can cause even the loss of positive definiteness of the covariance matrix. The alternative form, known as the ***Joseph form covariance update*** (see also (5.2.3-18)),

$$\begin{aligned} P(k|k, i) &= [I - W(k, i)h_i(k)']P(k|k, i-1)[I - W(k, i)h_i(k)']' \\ &\quad + W(k, i)r_i(k)W(k, i)' \end{aligned} \quad (7.3.2-13)$$

avoids this, but is computationally more expensive.

While the above form of the covariance update is better behaved numerically, it is still too sensitive to round-off errors. Further reduction of the effect of numerical errors can be obtained by using the square-root approach, which effectively doubles the numerical precision.

## 7.4 SQUARE-ROOT FILTERING

### 7.4.1 The Steps in Square-Root Filtering

The **square-root filtering** implementation of the KF carries out the covariance computations for the square root of the state covariance matrix, symbolically written as  $P^{1/2}$ , where

$$P = P^{1/2}(P^{1/2})' \triangleq \mathcal{P}\mathcal{P}' \quad (7.4.1-1)$$

The **square root of a matrix** is not unique, so several approaches are possible. The approach described in the sequel uses the  **$LDL'$  factorization** of a (positive definite) matrix

$$P = LDL' \quad (7.4.1-2)$$

where  $L$  is a **unit lower triangular matrix**, that is,

$$L_{ii} = 1 \quad \forall i \quad (7.4.1-3)$$

$$L_{ij} = 0 \quad i < j \quad (7.4.1-4)$$

and  $D$  is a **diagonal matrix with positive elements**.

The matrix square root  $\mathcal{P}$  in (7.4.1-1) corresponding to (7.4.1-2) is

$$\mathcal{P} = LD^{1/2} \triangleq \mathcal{L} \quad (7.4.1-5)$$

which is unique, except for the signs of the (diagonal) elements of  $D^{1/2}$ , which are taken as positive. Note that  $\mathcal{L}$  is a **lower triangular matrix**. Consequently, one has the following factorization

$$P = \mathcal{L}\mathcal{L}' \triangleq \mathcal{L}\mathcal{U} \quad (7.4.1-6)$$

where  $\mathcal{U}$  is an **upper triangular matrix**, which is known as the **the Cholesky factorization**.

The square-root filtering algorithm consists of the following steps:

1. Factorization of the initial covariance.
2. Computation of the factors of the predicted state covariance.
3. Computation of the factors of the updated state covariance and computation of the filter gain.

These steps are discussed in the next subsections.

### 7.4.2 The $LDL'$ Factorization

Given the positive definite  $n \times n$  matrix  $P$ , its decomposition, called the  **$LDL'$  factorization**,

$$P = [p_{ij}] = LDL' \quad (7.4.2-1)$$

where  $L$  is a *unit lower triangular matrix* and  $D$  is a *diagonal matrix with positive elements*, is obtained as follows [Bierman77]:

For  $j = 1, \dots, n - 1$

$$d_j = P_{jj}$$

$L_{jj} = 1$  (all others are zero)

For  $k = j + 1, \dots, n$

For  $i = k, \dots, n$

$$P_{ik} := P_{ik} - L_{ij}P_{kj}$$

$$L_{kj} = \frac{P_{kj}}{d_j}$$

$$d_n = P_{nn}$$

Note that since the diagonal terms of  $P$  have to be positive (because the matrix was assumed positive definite), the elements of the diagonal matrix  $D$  are guaranteed to be positive.

### 7.4.3 The Predicted State Covariance

The standard equation for the state prediction covariance is

$$P(k|k-1) = F(k-1)P(k-1|k-1)F(k-1)' + \Gamma(k-1)Q(k-1)\Gamma(k-1)' \quad (7.4.3-1)$$

or, in simpler notation, without time arguments

$$\bar{P} = FPF' + \Gamma Q \Gamma' \quad (7.4.3-2)$$

where  $P$  denotes the previous updated state covariance and  $\bar{P}$  denotes state prediction covariance.

Starting from the factorized form

$$P(k-1|k-1) \triangleq P = LDL' \quad (7.4.3-3)$$

we want to find the new factorized form

$$P(k|k-1) \triangleq \bar{P} = \bar{L}\bar{D}\bar{L}' \quad (7.4.3-4)$$

The brute force method would be to evaluate (7.4.3-2) and find its LDL' factorization using the method described in Subsection 7.4.2.

A better technique that takes advantage of the existing factorizations to obtain the ***factorized prediction covariance*** is as follows.

It is assumed that  $Q$  has been factorized as

$$Q = L_Q D_Q L_Q' \quad (7.4.3-5)$$

It will be shown how the factors of  $P$  given in (7.4.3-3) and the factors of  $Q$  given in (7.4.3-5) can be used to find the factors of  $\bar{P}$ .

Then we are looking for  $\bar{L}$ ,  $\bar{D}$  such that

$$\bar{P} = \bar{L} \bar{D} \bar{L}' = F L D L' F' + \Gamma L_Q D_Q L_Q' \Gamma' \quad (7.4.3-6)$$

Equation (7.4.3-6) can be rewritten as the product

$$\bar{P} = A A' \quad (7.4.3-7)$$

with the following  $n_x \times 2n_x$  matrix

$$A \triangleq \begin{bmatrix} F L D^{1/2} & \Gamma L_Q D_Q^{1/2} \end{bmatrix} \triangleq \begin{bmatrix} a_1' \\ a_2' \\ \vdots \\ a_{n_x}' \end{bmatrix} \triangleq \text{col}(a_i') \quad (7.4.3-8)$$

where  $a_i$  are  $2n_x$ -vectors.

Using the **Gram-Schmidt orthogonalization** procedure (described in Subsection 7.4.6) on the *rows* of the above matrix, one obtains

$$A = \bar{L} V \quad (7.4.3-9)$$

In the above,  $\bar{L}$  is a *unit lower triangular matrix* (of dimension  $n_x \times n_x$ ) and  $V$  (of dimension  $n_x \times 2n_x$ ) satisfies

$$V V' = \bar{D} \quad (7.4.3-10)$$

where  $\bar{D}$  is *diagonal with positive elements* (of dimension  $n_x \times n_x$ ) because  $V$  has orthogonal rows.

Thus

$$\bar{P} = A A' = \bar{L} V V' \bar{L}' = \bar{L} \bar{D} \bar{L}' \quad (7.4.3-11)$$

is the sought-after LDL' factorization of the prediction covariance (7.4.3-4).

Therefore, to obtain the factors  $\bar{L}$  and  $V$  of the prediction covariance  $\bar{P}$ , one carries out the Gram-Schmidt orthogonalization of the matrix  $A$  in (7.4.3-8), which contains the factors  $L$  and  $D$  of the previous updated state covariance  $P$  and the factors  $L_Q$  and  $D_Q$  of the process noise covariance  $Q$ .

### 7.4.4 The Filter Gain and the Updated State Covariance

Assuming a scalar measurement update, the implementation of the state update given in (7.3.2-5) requires the calculation of the gain  $W$ , which is an  $n_x$ -vector. This is obtained together with the factorized form of the updated state covariance as follows.

With the predicted state covariance factorized as  $\bar{L}\bar{D}\bar{L}'$ , we are looking for the factorized updated covariance

$$P \triangleq LDL' = \bar{L}\bar{D}\bar{L}' - \frac{\bar{L}\bar{D}\bar{L}'hh'\bar{L}\bar{D}\bar{L}'}{h'\bar{L}\bar{D}\bar{L}'h + r} \quad (7.4.4-1)$$

where the time arguments and subscripts have been omitted for simplicity.

Defining

$$f = \bar{L}'h = [f_1 \ \cdots \ f_n]' \quad (7.4.4-2)$$

Equation (7.4.4-1) can be written as

$$LDL' = \bar{L} \left[ \bar{D} - \frac{\bar{D}f(\bar{D}f)'}{f'\bar{D}f + r} \right] \bar{L}' \quad (7.4.4-3)$$

With

$$\bar{D} = \text{diag}[\bar{d}_1, \dots, \bar{d}_n] \quad (7.4.4-4)$$

one has

$$\bar{D}f = [\bar{d}_1 f_1 \ \cdots \ \bar{d}_n f_n]' \quad (7.4.4-5)$$

Denote the *columns* of  $\bar{L}$  as  $\bar{\ell}_i$  (which are  $n_x$ -vectors), that is,

$$\bar{L} = [\bar{\ell}_1 \ \cdots \ \bar{\ell}_n] \quad (7.4.4-6)$$

The **covariance update and gain calculation** algorithm consists of the following [Kleinman89]:

1. Initialize

$$\alpha_{n+1} = r \quad (7.4.4-7)$$

$$\alpha_n = r + \bar{d}_n f_n^2 \quad (7.4.4-8)$$

$$\xi = [0 \ \cdots \ 0 \ \bar{d}_n f_n]' \quad (7.4.4-9)$$

$$d_n = \frac{\alpha_{n+1}}{\alpha_n} \bar{d}_n \quad (7.4.4-10)$$

2. For  $i = n-1, \dots, 1$

$$\beta = -\frac{f_i}{\alpha_{i+1}} \quad (7.4.4-11)$$

$$\alpha_i = \alpha_{i+1} + \bar{d}_i f_i^2 \quad (7.4.4-12)$$

$$d_i = \frac{\alpha_{i+1}}{\alpha_i} \bar{d}_i \quad (7.4.4-13)$$

$$\ell_i = \bar{\ell}_i + \beta \xi \quad (7.4.4-14)$$

$$\xi := \xi + \bar{\ell}_i d_i f_i \quad (7.4.4-15)$$

### 3. Gain vector for the state update

$$W = \frac{1}{\alpha_1} \xi \quad (7.4.4-16)$$

In the above  $d_i$ ,  $i = 1, \dots, n_x$ , are the elements of the diagonal matrix  $D$  and the  $n_x$ -vectors  $\ell_i$ ,  $i = 1, \dots, n_x$ , are the columns of the unit lower diagonal matrix  $L$ . These two matrices define the LDL' factorization (7.4.4-1) of the updated state covariance matrix  $P$ .

## 7.4.5 Overview of the Square-Root Sequential Scalar Update Algorithm

It is assumed that the measurement noise covariance matrix is diagonal. If this is not the case, a Cholesky factorization is performed as in Subsection 7.4.2 and the transformations (7.3.2-10) to (7.3.2-12) are carried out.

Figure 7.4.5-1 describes the sequence of computations for the square-root sequential scalar update algorithm. The last block is repeated for each component of the  $n_z$ -dimensional measurement vector.

## 7.4.6 The Gram-Schmidt Orthogonalization Procedure

Given  $N$ -vectors  $a_i$ ,  $i = 1, \dots, n$ , where  $N \geq n$ , assumed to be linearly independent, let

$$v_i \triangleq a_i - \sum_{j=1}^{i-1} \frac{a_i' v_j}{v_j' v_j} v_j \quad i = 1, \dots, n \quad (7.4.6-1)$$

Then the vectors  $v_i$ ,  $i = 1, \dots, n$ , form an orthogonal set. Equation (7.4.6-1) can be rewritten as

$$a_i = v_i + \sum_{j=1}^{i-1} \bar{L}_{ij} v_j \quad (7.4.6-2)$$

where

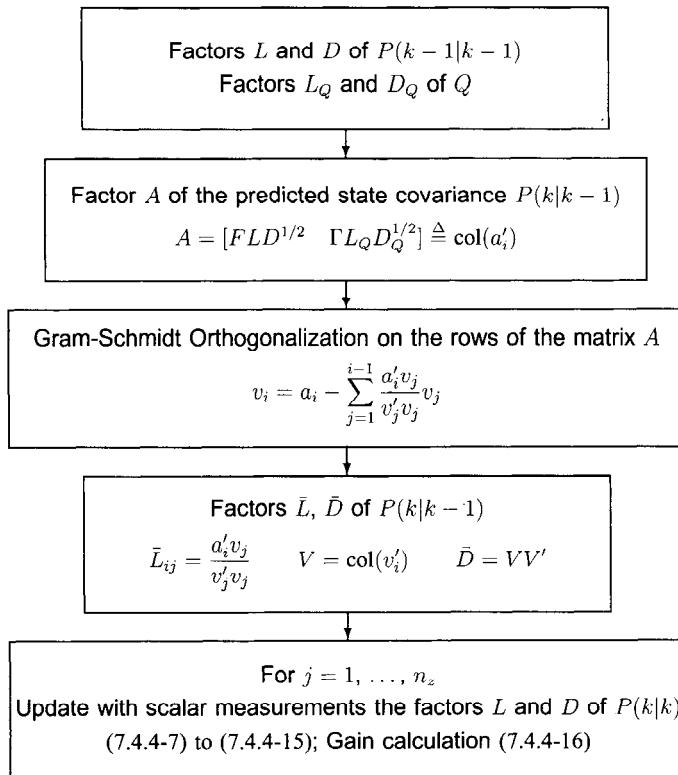
$$\bar{L}_{ij} \triangleq \frac{a_i' v_j}{v_j' v_j} \quad (7.4.6-3)$$

Then

$$A = \begin{bmatrix} a_1' \\ a_2' \\ \vdots \\ a_n' \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \bar{L}_{21} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \bar{L}_{n1} & \bar{L}_{n2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} v_1' \\ v_2' \\ \vdots \\ v_n' \end{bmatrix} \triangleq \bar{L}V \quad (7.4.6-4)$$

where  $\bar{L}$  is a unit lower triangular matrix ( $n \times n$ ) and

$$V \triangleq \begin{bmatrix} v_1' \\ v_2' \\ \vdots \\ v_n' \end{bmatrix} \triangleq \text{col}(v_i') \quad (7.4.6-5)$$



**Figure 7.4.5-1:** The square-root sequential covariance update algorithm.

Note that the transpose of  $v_i$  is the *i*th *row* of  $V$ .

From the orthogonality of the  $n$ -vectors  $v_i$ ,  $i = 1, \dots, n$ , it follows that

$$VV' = \bar{D} \quad (7.4.6-6)$$

is a diagonal matrix (also  $n \times n$ ).

Equation (7.4.6-1) is called the **Gram-Schmidt orthogonalization procedure**.

The so-called modified Gram-Schmidt orthogonalization procedure (e.g., [Bierman77]) has improved numerical properties compared to the standard procedure described above.

### Application to the Generation of Correlated Random Variables

This orthogonalization procedure can also be used in the reverse direction to provide a recursion for the **generation of correlated random variables** from a standard random number generator that yields independent random variables.

Similarly, one can obtain a simple recursion for the **generation of a random sequence with exponential autocorrelation** (see problem 7-1).

These techniques are useful in simulations that require the generation of correlated random variables.

## 7.5 NOTES AND PROBLEMS

### 7.5.1 Bibliographical Notes

The most comprehensive documentation of numerical algorithms for linear estimation can be found in [Bierman77, Maybeck79, Maybeck82]. The information filter and its use for estimation start-up without initial estimates is discussed in [Maybeck79]. Applications of the information filter to robotics are discussed in [Mutambara98]. The material presented in this chapter is based in part on private communications from D. L. Kleinman and K. R. Pattipati.

A discussion of the use of parallel computers for the implementation of linear estimation can be found in [O'Halloran88]. Parallel implementation of factorized (square-root) estimation is discussed in [Itzkowitz89].

Several applications of covariance factorization in estimation as well as additional references on this topic can be found in [Baheti90].

### 7.5.2 Problems

- 7-1 Generation of a random sequence with exponential autocorrelation.** Given a set of independent zero-mean unit-variance random variables,  $x_i$ ,  $i = 1, \dots$ , find a *recursion*

that yields the zero-mean sequence  $y_i$ ,  $i = 1, \dots$ , with the autocorrelation function

$$E[y_i y_j] = \rho^{|i-j|} \quad |\rho| < 1$$

*Hint:* The form of the recursion should be  $y_{i+1} = f(y_i, x_{i+1})$ .

- 7-2 Start-up of the filter for the DWNA model without initial estimates.** For the DWNA model (Subsection 6.3.2) derive  $\hat{x}(2)$  and  $P(2|2)$  starting from  $P(0|0) = 0$ .
- 7-3 Start-up of the filter for the CWNA model without initial estimates.** For the CWNA model (Subsection 6.2.2) derive  $\hat{x}(2)$  and  $P(2|2)$  starting from  $P(0|0) = 0$ .
- 7-4 Start-up of the filter for the DWPA model without initial estimates.** For the DWPA model (Subsection 6.3.3) derive  $\hat{x}(3)$  and  $P(3|3)$  starting from  $P(0|0) = 0$ .
- 7-5 Start-up of the filter for the CWPA model without initial estimates.** For the CWPA model (Subsection 6.2.3) derive  $\hat{x}(3)$  and  $P(3|3)$  starting from  $P(0|0) = 0$ .

# Chapter 8

## EXTENSIONS OF DISCRETE-TIME LINEAR ESTIMATION

### 8.1 INTRODUCTION

#### 8.1.1 Outline

The major assumptions of the optimal linear estimator for dynamic systems (the Kalman filter) are as follows:

1. The process noise sequence is white.
2. The measurement noise sequence is white.
3. The two noise sequences are uncorrelated.

The next three sections discuss the procedures to reduce the problems where the above assumptions are not satisfied, to the standard case.

Section 8.2 presents the *prewhitening* and *state augmentation* technique needed in the case of *autocorrelated process noise*. Section 8.3 deals with the case of *correlated noise sequences* and shows how this correlation can be eliminated. Section 8.4 considers the situation of *autocorrelated measurement noise*, which is solved with the help of the techniques from the previous two sections.

Estimation of the state at times other than the current time is discussed in the following two sections. *Prediction* is treated in Section 8.5 and *smoothing* — estimation of the state at an earlier time than the last data point — is the topic of Section 8.6.

#### 8.1.2 Extensions of Estimation — Summary of Objectives

Reduce the problems of state estimation in the following nonstandard situations:

- Autocorrelated process noise

- Cross-correlated process and measurement noise
- Autocorrelated measurement noise

to the standard problem where both noise sequences are *white and mutually uncorrelated*.

Provide equations for estimation of the state at times other than the time of the latest measurement:

- Prediction — beyond the data interval
- Smoothing — within the data interval

## 8.2 AUTOCORRELATED PROCESS NOISE

### 8.2.1 The Autocorrelated Process Noise Problem

Consider the dynamic system driven by an *autocorrelated process noise* (also called *colored noise*)

$$x(k+1) = Fx(k) + v_c(k) \quad (8.2.1-1)$$

where  $v_c(k)$  is a zero-mean stationary but not white sequence

$$E[v_c(k)v_c(j)'] = Q(k-j) \quad (8.2.1-2)$$

Then the state  $x(k)$  of the above system is *not a Markov sequence*. The goal is to estimate the state from the measurements

$$z(k) = Hx(k) + w(k) \quad (8.2.1-3)$$

where

$$E[w(k)w(j)'] = R\delta_{kj} \quad (8.2.1-4)$$

$$E[v(k)w(j)'] = 0 \quad (8.2.1-5)$$

that is, with white measurement noise uncorrelated from the process noise.

In order to be able to apply the standard estimation results, one has to reformulate the problem into one with a state that is a Markov sequence — that is, one where the process noise driving the dynamic system is white.

This is accomplished by obtaining the *prewhitening system* (or *shaping filter*) for the process noise and attaching it to the original system as illustrated in Subsection 1.4.21. This amounts to carrying out a *state augmentation* such that the augmented state becomes a Markov sequence.

This is illustrated in the sequel with an example motivated by maneuvering targets.

## 8.2.2 An Exponentially Autocorrelated Noise

### The Acceleration Model

In Chapter 6 a second-order kinematic model for target tracking was obtained modeling the acceleration as *white noise*. In the approach discussed here the target acceleration  $a(t)$  is modeled as an **exponentially autocorrelated noise** with mean zero and

$$R(\tau) = E[a(t)a(t + \tau)] = \sigma_m^2 e^{-\alpha|\tau|} \quad \alpha > 0 \quad (8.2.2-1)$$

This is also known as an **Ornstein-Uhlenbeck process**. In the above,  $\sigma_m^2$  is the **instantaneous variance** of the acceleration and  $1/\alpha$  is the time constant of the target acceleration autocorrelation, that is, the **decorrelation time** is approximately  $2/\alpha$ .

### The Noise Variance and the Target Physical Parameters

The variance of the (instantaneous) acceleration can be obtained by assuming it to be

1. Equal to a maximum value  $a_M$  with probability  $p_M$  and  $-a_M$  with the same probability (chosen by the designer)
2. Equal to zero with probability  $p_0$  (also a design parameter)
3. Uniformly distributed in  $[-a_M, a_M]$  with the remaining probability mass

This results in the following mixed pmf/pdf (a combination of three point masses and a uniform pdf, with total probability mass unity)

$$p(a) = [\delta(a - a_M) + \delta(a + a_M)]p_M + \delta(a)p_0 + [1(a + a_M) - 1(a - a_M)] \frac{1 - p_0 - 2p_M}{2a_M} \quad (8.2.2-2)$$

where  $1(\cdot)$  denotes the unit step function.

It can be easily shown that the variance corresponding to (8.2.2-2) is

$$\sigma_m^2 = \frac{a_M^2}{3}(1 + 4p_M - p_0) \quad (8.2.2-3)$$

The dynamic model — the **prewhitening system** — that yields (8.2.2-1) is the first-order Markov process

$$\dot{a}(t) = -\alpha a(t) + \tilde{v}(t) \quad (8.2.2-4)$$

driven by the white noise  $\tilde{v}(t)$  with

$$E[\tilde{v}(t)\tilde{v}(\tau)] = 2\alpha\sigma_m^2\delta(t - \tau) \quad (8.2.2-5)$$

This will be used in the next subsection as a subsystem to augment the state equations according to the procedure discussed in Subsection 1.4.21.

## Remarks

1. As  $\alpha$  increases, the process  $a(t)$  becomes uncorrelated faster. At the limit (suitably defined — see problem 8-4), as  $\alpha \rightarrow \infty$  and  $\sigma_m \rightarrow \infty$ , the acceleration becomes white noise, that is,

$$\lim_{\alpha \rightarrow \infty, \sigma_m \rightarrow \infty} R(\tau) = q\delta(\tau) \quad (8.2.2-6)$$

The relationship between  $q$  in (8.2.2-6) and the parameters  $\sigma_m$  and  $\alpha$  is

$$\frac{2}{\alpha} \sigma_m^2 = q \quad (8.2.2-7)$$

This case corresponds to the white noise acceleration model (WNA), which is of second order, discussed in Subsection 6.2.2.

2. For  $\alpha \rightarrow 0$ , (8.2.2-4) yields  $a(t)$  as the integral of white noise, that is, the acceleration becomes a *Wiener process*, which is not stationary (note that the process defined in (8.2.2-1) is stationary). This is suitable for maneuver modeling for time-varying acceleration with a suitable value for the process noise variance  $q$ .

This case corresponds to the Wiener process acceleration (WPA) model, which is of third order, and was presented in Subsection 6.2.3.

### 8.2.3 The Augmented State Equations

With the acceleration a white noise, the state would consist of position and velocity (per coordinate), as in Subsection 6.2.2. Augmented with the acceleration, the state for the generic coordinate  $\xi$  is denoted as

$$x = [\xi \ \dot{\xi} \ \ddot{\xi}]' \quad (8.2.3-1)$$

where  $\ddot{\xi} = a$  is the acceleration.

The continuous-time state equations are

$$\dot{x}_1(t) = x_2(t) \quad (8.2.3-2)$$

$$\dot{x}_2(t) = x_3(t) \quad (8.2.3-3)$$

to which one attaches the prewhitening subsystem (of dimension 1 in this case — in general it can be of higher dimension)

$$\dot{x}_3(t) = -\alpha x_3(t) + \tilde{v}(t) \quad (8.2.3-4)$$

The **augmented state equation** in vector form is then

$$\dot{x}(t) = Ax(t) + D\tilde{v}(t) \quad (8.2.3-5)$$

with the system matrix

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\alpha \end{bmatrix} \quad (8.2.3-6)$$

and  $\tilde{v}(t)$  is the (scalar) white process noise from (8.2.2-4), which enters the system through the gain

$$D = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (8.2.3-7)$$

Note that  $\alpha = 0$  makes the above identical to (6.2.3-4).

The discrete-time equation corresponding to (8.2.3-5) for sampling period  $T$  is

$$x(k+1) = Fx(k) + v(k) \quad (8.2.3-8)$$

where

$$F = e^{AT} = \begin{bmatrix} 1 & T & (\alpha T - 1 + e^{-\alpha T})/\alpha^2 \\ 0 & 1 & (1 - e^{-\alpha T})/\alpha \\ 0 & 0 & e^{-\alpha T} \end{bmatrix} \quad (8.2.3-9)$$

The discrete-time process noise  $v(k)$  has the covariance matrix

$$Q = 2\alpha\sigma_m^2 \begin{bmatrix} T^5/20 & T^4/8 & T^3/6 \\ T^4/8 & T^3/3 & T^2/2 \\ T^3/6 & T^2/2 & T \end{bmatrix} \quad (8.2.3-10)$$

Expression (8.2.3-10) assumes

$$\alpha T \ll 1 \quad (8.2.3-11)$$

that is, that the sampling time  $T$  is much less than the time constant  $1/\alpha$  of the maneuver autocorrelation. If this is not the case, then the sampling is too slow to "catch" the autocorrelation in the acceleration. Thus, if one uses this model, then (8.2.3-11) should be satisfied and (8.2.3-10) is adequate.

Typical values of  $1/\alpha$  for aircraft are 20 s for a slow turn and 5 s for an evasive maneuver; atmospheric turbulence has a time constant around 1 s.

The exact expression of  $Q$  has the following elements

$$q_{11} = \frac{\sigma_m^2}{\alpha^4} \left[ 1 - e^{-2\alpha T} + 2\alpha T + \frac{2\alpha^3 T^3}{3} - 2\alpha^2 T^2 - 4\alpha T e^{-\alpha T} \right] \quad (8.2.3-12)$$

$$q_{12} = \frac{\sigma_m^2}{\alpha^3} [e^{-2\alpha T} + 1 - 2e^{-\alpha T} + 2\alpha T e^{-\alpha T} - 2\alpha T + \alpha^2 T^2] \quad (8.2.3-13)$$

$$q_{13} = \frac{\sigma_m^2}{\alpha^2} [1 - e^{-2\alpha T} - 2\alpha T e^{-\alpha T}] \quad (8.2.3-14)$$

$$q_{22} = \frac{\sigma_m^2}{\alpha^2} [4e^{-\alpha T} - 3 - e^{-2\alpha T} + 2\alpha T] \quad (8.2.3-15)$$

$$q_{23} = \frac{\sigma_m^2}{\alpha} [e^{-2\alpha T} + 1 - 2e^{-\alpha T}] \quad (8.2.3-16)$$

$$q_{33} = \sigma_m^2 [1 - e^{-2\alpha T}] \quad (8.2.3-17)$$

### 8.2.4 Estimation with Autocorrelated Process Noise — Summary

To estimate the state of a system driven by an *autocorrelated process noise*,

- The prewhitening subsystem for the autocorrelated process noise has to be obtained.
- The original system is augmented with this subsystem such that the resulting system is driven by white noise.

This has been illustrated with an application from maneuvering targets. Target maneuvers, which by their nature have a certain duration in a certain direction, can be modeled by autocorrelated noise.

An exponentially decaying autocorrelation for target maneuvers can model

- The *magnitude of the maneuvers*
- The *decorrelation time* of the maneuvers

The magnitude of the maneuvers is modeled by their instantaneous variance. One possible procedure to obtain it, based on

- The maximum maneuver and the probability that it takes place
- The probability that no maneuver takes place
- A random maneuver of magnitude up to the maximum

has been illustrated.

The time constant of the decorrelation of the maneuvers depends on the target. This exponentially autocorrelated acceleration is obtained from a subsystem that is a first-order Markov process driven by white noise.

Higher-order prewhitening subsystems can be used in a similar manner. The transfer functions of such systems can be obtained by factorization of the power spectral density of its output (the colored noise) as discussed briefly in Subsection 1.4.21. For more general techniques, see, e.g., [Poor88].

## 8.3 CROSS-CORRELATED MEASUREMENT AND PROCESS NOISE

### 8.3.1 Cross-Correlation at the Same Time

Consider the model (time-invariant for the sake of notational simplicity)

$$x(k+1) = Fx(k) + v(k) \quad (8.3.1-1)$$

$$z(k) = Hx(k) + w(k) \quad (8.3.1-2)$$

$$E[v(k)v(j)'] = Q\delta_{kj} \quad (8.3.1-3)$$

$$E[w(k)w(j)'] = R\delta_{kj} \quad (8.3.1-4)$$

$$E[v(k)w(j)'] = U\delta_{kj} \quad (8.3.1-5)$$

where the last equation indicates ***cross-correlated noise sequences*** — a nonzero cross-correlation between the process and measurement noise sequences, that is, it is *not a standard filtering problem*. Note that the cross-correlation is between  $v(k)$  and  $w(k)$ , i.e., only if the time arguments are the same.

The plant equation can be rewritten such that it has a new process noise uncorrelated with the measurement noise. Using an *arbitrary matrix*  $T$ , to be determined later, one can write

$$\begin{aligned} x(k+1) &= Fx(k) + v(k) + T[z(k) - Hx(k) - w(k)] \\ &= (F - TH)x(k) + v(k) - Tw(k) + Tz(k) \end{aligned} \quad (8.3.1-6)$$

Note that (8.3.1-6) is *completely equivalent* to (8.3.1-1) since they differ only by a term that is identically zero.

Denote the new transition matrix as

$$F^* \triangleq F - TH \quad (8.3.1-7)$$

the new process noise as

$$v^*(k) \triangleq v(k) - Tw(k) \quad (8.3.1-8)$$

and the last term in (8.3.1-6), which is a *known input*, as

$$u^*(k) \triangleq Tz(k) \quad (8.3.1-9)$$

Then, one can write the modified state equation (8.3.1-6) as

$$x(k+1) = F^*x(k) + v^*(k) + u^*(k) \quad (8.3.1-10)$$

Setting the cross-correlation between the new process noise and the measurement noise to zero

$$E[v^*(k)w(k)'] = E[[v(k) - Tw(k)]w(k)'] = U - TR = 0 \quad (8.3.1-11)$$

yields the unspecified matrix  $T$  as

$$T = UR^{-1} \quad (8.3.1-12)$$

With the above, the covariance of the new process noise

$$\begin{aligned} Q^* &\triangleq E[v^*(k)v^*(k)'] \\ &= E[[v(k) - UR^{-1}w(k)][v(k) - UR^{-1}w(k)']] \end{aligned} \quad (8.3.1-13)$$

is obtained as

$$Q^* = Q - UR^{-1}U' \quad (8.3.1-14)$$

Using (8.3.1-7), (8.3.1-9), and (8.3.1-12), the modified state equation (8.3.1-10) can be rewritten as

$$x(k+1) = (F - UR^{-1}H)x(k) + v^*(k) + UR^{-1}z(k) \quad (8.3.1-15)$$

where the known input — the last term above — is written out explicitly.

### 8.3.2 Cross-Correlation One Time Step Apart

Consider the same model as before in (8.3.1-1)–(8.3.1-4), but with (8.3.1-5) replaced by

$$E[v(k)w(j)'] = U\delta_{k,j-1} \quad (8.3.2-1)$$

i.e., with the cross-correlation nonzero only between  $v(k)$  and  $w(k+1)$ .

Even though this is not a standard problem, one can use the derivation of the standard filter from Subsection 5.2.3 with appropriate modifications to obtain the filter for this case.

The state prediction error is

$$\tilde{x}(k+1|k) \triangleq x(k+1) - \hat{x}(k+1|k) = F\tilde{x}(k|k) + v(k) \quad (8.3.2-2)$$

With (8.3.2-1), the covariance between the state prediction error and the measurement noise is

$$E[\tilde{x}(k+1|k)w(k+1)'] = E[[F\tilde{x}(k|k) + v(k)]w(k+1)'] = U \quad (8.3.2-3)$$

The covariance between the state and the measurement (5.2.3-10) becomes

$$\begin{aligned} E[\tilde{x}(k+1|k)\tilde{z}(k+1|k)'|Z^k] &= E[\tilde{x}(k+1|k)[H\tilde{x}(k+1|k) + w(k+1)']'|Z^k] \\ &= P(k+1|k)H' + U \end{aligned} \quad (8.3.2-4)$$

The measurement prediction covariance (5.2.3-9) becomes

$$\begin{aligned} S(k+1) &= E[\tilde{z}(k+1|k)\tilde{z}(k+1|k)'] \\ &= E[[H\tilde{x}(k+1|k) + w(k+1)][H\tilde{x}(k+1|k) + w(k+1)']] \\ &= HP(k+1|k)H' + R + HU + U'H' \end{aligned} \quad (8.3.2-5)$$

The resulting *modified* filter gain is, using (8.3.2-4) and (8.3.2-5),

$$W(k+1) = [P(k+1|k)H' + U][HP(k+1|k)H' + R + HU + U'H']^{-1} \quad (8.3.2-6)$$

Except for the above modified filter gain, all the filter equations are the same as in the standard case.

### Remark

Since  $v(k)$  is the process noise corresponding to the interval  $[t_k, t_{k+1}]$ , and  $w(j)$  is the measurement noise at  $t_j$ , it can be seen that in the previous Subsection the process noise is correlated with the measurement noise at the beginning of the above interval, while in the present Subsection it is correlated with the one at the end of the interval.

### 8.3.3 State Estimation with Decorrelated Noise Sequences — Summary

The state estimation problem with cross-correlated process and measurement noise sequences *at the same discrete time* has been transformed into the following equivalent problem defined by

- The modified state equation (8.3.1-15)
  - with covariance of the new process noise (8.3.1-14) and
  - with a known input
- The original measurement equation (8.3.1-2)
  - with the measurement noise covariance given by (8.3.1-4)

This is now a *standard problem* since the two noise sequences are uncorrelated according to (8.3.1-11).

The state estimation problem with cross-correlated process and measurement noise sequences *at one time step apart* can be handled in the same manner as the standard problem except that the expression of the filter gain is to be modified to account for this cross-correlation.

## 8.4 AUTOCORRELATED MEASUREMENT NOISE

### 8.4.1 Whitening of the Measurement Noise

Consider the model

$$x(k+1) = Fx(k) + v(k) \quad (8.4.1-1)$$

$$z(k) = Hx(k) + w_c(k) \quad (8.4.1-2)$$

$$w_c(k+1) = F_c w_c(k) + w_w(k) \quad (8.4.1-3)$$

$$E[w_w(k)w_w(j)'] = R_w \delta_{kj} \quad (8.4.1-4)$$

$$E[v(k)v(j)'] = Q \delta_{kj} \quad (8.4.1-5)$$

$$E[v(k)w_w(j)'] = 0 \quad (8.4.1-6)$$

where (8.4.1-3) indicates a Markov **autocorrelated measurement noise**.<sup>1</sup>

The “brute force” solution is to augment the state with  $w_c$ . This reduces the problem to a standard one, but with “perfect state observations,” which yields a singular state covariance — an undesirable feature.

To avoid this, introduce the following **differenced measurement**:

$$\begin{aligned} y(k) &\triangleq z(k+1) - F_c z(k) = Hx(k+1) + w_c(k+1) - F_c Hx(k) - F_c w_c(k) \\ &= Hv(k) + w_w(k) - F_c Hx(k) \end{aligned} \quad (8.4.1-7)$$

<sup>1</sup>This type of state equation implies that the ( $z$ -domain) transfer function has (stable) poles and also (possibly) zeros. If the transfer function has only zeros, the system cannot be represented by (8.4.1-3). In that case, the method to be presented in the sequel will not work — a different approach (which works) can be found in [Bar-Shalom95], Subsection 9.4.5.

which can be written as

$$y(k) = H^*x(k) + w(k) \quad (8.4.1-8)$$

where the new measurement matrix is

$$H^* \triangleq HF - F_c H \quad (8.4.1-9)$$

The new measurement noise

$$w(k) \triangleq Hv(k) + w_w(k) \quad (8.4.1-10)$$

is *white* but *correlated with the process noise*

$$E[w(k)w(j)'] = (HQH' + R_w)\delta_{kj} \triangleq R\delta_{kj} \quad (8.4.1-11)$$

$$E[v(k)w(k)'] = QH' \quad (8.4.1-12)$$

The technique from the previous section will be used to eliminate the cross-correlation between the two noise sequences. Rewrite the plant equation:

$$\begin{aligned} x(k+1) &= Fx(k) + v(k) + T[y(k) - H^*x(k) - w(k)] \\ &= (F - TH^*)x(k) + v(k) - Tw(k) + Ty(k) \end{aligned} \quad (8.4.1-13)$$

Define the new process noise

$$v^*(k) \triangleq v(k) - Tw(k) \quad (8.4.1-14)$$

and new transition matrix

$$F^* = F - TH^* \quad (8.4.1-15)$$

With this, one can write the new state equation as

$$x(k+1) = F^*x(k) + v^*(k) + Ty(k) \quad (8.4.1-16)$$

In the above,  $T$  will be chosen such that *the cross-correlation is zero*:

$$E[v^*(k)w(k)'] = E[[v(k) - Tw(k)]w(k)'] = QH' - TR = 0 \quad (8.4.1-17)$$

This yields

$$T = QH'R^{-1} = QH'(HQH' + R_w)^{-1} \quad (8.4.1-18)$$

The new transition matrix is then

$$F^* \triangleq F - TH^* = F - QH'(HQH' + R_w)^{-1}(HF - F_c H) \quad (8.4.1-19)$$

The new process noise covariance is

$$\begin{aligned} E[v^*(k)v^*(k)'] &= E[[v(k) - QH'R^{-1}w(k)][v(k) - QH'R^{-1}w(k)']] \\ &= Q - QH'R^{-1}HQ \triangleq Q^* \end{aligned} \quad (8.4.1-20)$$

The estimation can now be done with

- The state equation (8.4.1-16) with white process noise  $v^*(k)$  with covariance (8.4.1-20)
- The measurements (8.4.1-8), with white measurement noise  $w(k)$  with covariance (8.4.1-11)

These two noise sequences are mutually uncorrelated according to (8.4.1-17). This is a standard problem except that  $y(k)$  contains  $z(k+1)$ .

### 8.4.2 The Estimation Algorithm with the Whitened Measurement Noise

The estimation algorithm is described next. First note that, in view of (8.4.1-7),  $Y^k$  is equivalent to  $Z^{k+1}$ . Therefore,

$$\hat{x}(k|Y^k) = \hat{x}(k|Z^{k+1}) = \hat{x}(k|k+1) \quad (8.4.2-1)$$

$$P(k|Y^k) = P(k|Z^{k+1}) = P(k|k+1) \quad (8.4.2-1)$$

$$\hat{x}(k+1|Y^k) = \hat{x}(k+1|Z^{k+1}) = \hat{x}(k+1|k+1) \quad (8.4.2-2)$$

$$P(k+1|Y^k) = P(k+1|Z^{k+1}) = P(k+1|k+1) \quad (8.4.2-2)$$

The filter stages below are labeled according to the indices of  $x$  and  $y$ .

“Prediction” — conditional expectation of (8.4.1-16) given  $Y^k$ :

$$\begin{aligned} \hat{x}(k+1|Y^k) &= F^* \hat{x}(k|Y^k) + Ty(k) = F^* \hat{x}(k|k+1) + Ty(k) \\ &= \hat{x}(k+1|k+1) \end{aligned} \quad (8.4.2-3)$$

$$\begin{aligned} P(k+1|Y^k) &= F^* P(k|Y^k) F^{*\prime} + Q^* = F^* P(k|k+1) F^{*\prime} + Q^* \\ &= P(k+1|k+1) \end{aligned} \quad (8.4.2-4)$$

“Update”:

$$\begin{aligned} W(k+1) &= P(k+1|Y^k) H^{*\prime} S(k+1)^{-1} \\ &= P(k+1|k+1) H^{*\prime} S(k+1)^{-1} \end{aligned} \quad (8.4.2-5)$$

$$\begin{aligned} S(k+1) &= H^* P(k+1|Y^k) H^{*\prime} + R \\ &= H^* P(k+1|k+1) H^{*\prime} + R \end{aligned} \quad (8.4.2-6)$$

$$\begin{aligned} \hat{x}(k+1|Y^{k+1}) &= \hat{x}(k+1|Y^k) + W(k+1)[y(k+1) - H^* \hat{x}(k+1|Y^k)] \\ &= \hat{x}(k+1|k+1) + W(k+1)[y(k+1) - H^* \hat{x}(k+1|k+1)] \\ &= \hat{x}(k+1|k+2) \end{aligned} \quad (8.4.2-7)$$

$$\begin{aligned} P(k+1|Y^{k+1}) &= P(k+1|Y^k) - W(k+1) S(k+1) W(k+1)' \\ &= P(k+1|k+1) - W(k+1) S(k+1) W(k+1)' \\ &= P(k+1|k+2) \end{aligned} \quad (8.4.2-8)$$

The initiation is done starting with the initial estimate  $\hat{x}(0|0)$  with covariance  $P(0|0)$  and then using (8.4.2-7) and (8.4.2-8) with  $k = -1$ . Following this, the “prediction” stage is carried out and the cycle continues with the “update.”

### 8.4.3 Autocorrelated Measurement Noise — Summary

The procedure to handle Markov autocorrelated measurement noise is as follows:

- The measurements are differenced such that they result in a new measurement sequence with white noise.
- Since this new measurement noise is cross-correlated with the process noise, the technique to eliminate the cross-correlation between the two noise sequences is used.
- Another consequence of the measurement differencing is the lag by one time step in the new measurements — this is handled by letting a one-step prediction of the state become the latest estimate.

## 8.5 PREDICTION

### 8.5.1 Types of Prediction

*Prediction*<sup>2</sup> is the *estimation of the state at time  $j$  beyond the data interval*, that is, based on data up to an earlier time  $k < j$ ,

$$\hat{x}(j|k) = E[x(j)|Z^k] \quad (8.5.1-1)$$

There are several types of prediction:

- (a) **Fixed-point prediction:** This is done to the fixed point in time  $j = N$  (e.g., an intercept time) based on the data up to time  $k$ , where  $k$  is changing.
- (b) **Fixed-lead prediction:** This is done for a fixed number of  $L$  steps ahead, that is,  $j = k + L$ , where  $k$  varies and  $L$  is the lead.
- (c) **Fixed-interval prediction:** This is done based on the data from the fixed interval up to  $k = N$  to times beyond it,  $j = N + 1, \dots$ .

These are illustrated in Fig. 8.5.1-1.

The equations for these three types of prediction are given next.

### 8.5.2 The Algorithms for the Different Types of Prediction

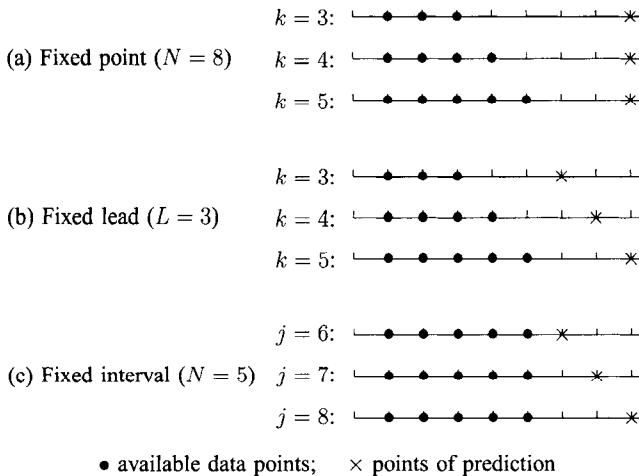
#### Fixed-Point Prediction

Based on (4.3.3-1), the *fixed-point prediction* of the state to the fixed time (point)  $N$  from time  $k$  is

$$\begin{aligned} \hat{x}(N|k) &= E \left[ \left[ \prod_{j=0}^{N-k-1} F(N-1-j) \right] x(k) + \sum_{i=k}^{N-1} \left[ \prod_{j=0}^{N-i-2} F(N-1-j) \right] v(i) | Z^k \right] \\ &= \left[ \prod_{j=0}^{N-k-1} F(N-1-j) \right] \hat{x}(k|k) \end{aligned} \quad (8.5.2-1)$$

---

<sup>2</sup>It is said that this is a rather difficult enterprise, especially when it is about the future.



**Figure 8.5.1-1:** The three types of prediction.

since the process noise, being zero mean and white, yields zero after the expectation is applied.

The above can be rewritten as

$$\hat{x}(N|k) = \left[ \prod_{j=0}^{N-k-1} F(N-1-j) \right] [F(k-1)\hat{x}(k-1|k-1) + W(k)\nu(k)] \quad (8.5.2-2)$$

which becomes the recursion

$$\hat{x}(N|k) = \hat{x}(N|k-1) + \left[ \prod_{j=0}^{N-k-1} F(N-1-j) \right] W(k)\nu(k) \quad (8.5.2-3)$$

It can be shown (see problem 8-1) that the expected value of the prediction to  $N$  from  $i$ , conditioned on the data through time  $k$ , is

$$E[\hat{x}(N|i)|Z^k] = \hat{x}(N|k) \quad \forall i > k \quad (8.5.2-4)$$

This quantity is the *predicted value at  $k$  of the prediction to be available at the later time  $i$* .

The covariance of this “prediction of a prediction” is

$$E[(\hat{x}(N|i) - \hat{x}(N|k))(\hat{x}(N|i) - \hat{x}(N|k))'|Z^k] = P(N|k) - P(N|i) \quad \forall i > k \quad (8.5.2-5)$$

### Fixed-Lead Prediction

Similarly to (8.5.2-1), one has the ***fixed-lead prediction*** for the fixed lead of  $L$  steps, given by

$$\begin{aligned}\hat{x}(k+L|k) &= \left[ \prod_{j=0}^{L-1} F(k+L-1-j) \right] \hat{x}(k|k) \\ &= \left[ \prod_{j=0}^{L-1} F(k+L-1-j) \right] [F(k-1)\hat{x}(k-1|k-1) + W(k)\nu(k)]\end{aligned}\quad (8.5.2-6)$$

which becomes the recursion

$$\begin{aligned}\hat{x}(k+L|k) &= F(k+L-1)\hat{x}(k+L-1|k-1) \\ &\quad + \left[ \prod_{j=0}^{L-1} F(k+L-1-j) \right] W(k)\nu(k)\end{aligned}\quad (8.5.2-7)$$

The covariance associated with the above follows from the standard propagation equation

$$P(k+l|k) = F(k+l-1)P(k+l-1|k)F(k+l-1)' + Q(k+l-1) \quad l = 1, \dots, L \quad (8.5.2-8)$$

### Fixed-Interval Prediction

The propagation equation of the ***fixed-interval prediction*** of the state at time  $j$  based on the fixed data interval up to  $N$  is

$$\hat{x}(j|N) = F(j-1)\hat{x}(j-1|N) \quad (8.5.2-9)$$

or

$$\begin{aligned}\hat{x}(j|N) &= F(j-1)\hat{x}(j-1|N) \\ &= \left[ \prod_{l=0}^{j-N-1} F(j-N-1-l) \right] \hat{x}(N|N) \quad j > N\end{aligned}\quad (8.5.2-10)$$

The associated covariance follows from the standard state prediction covariance propagation equation

$$P(l|N) = F(l-1)P(l-1|N)F(l-1)' + Q(l-1) \quad l = N+1, \dots, j \quad (8.5.2-11)$$

### 8.5.3 Prediction — Summary

Prediction is the estimation of the state at time beyond the data interval.

There are several types of prediction:

- Fixed-point prediction: to a fixed future time based on the data up to the current time, which varies.
- Fixed-lead prediction: for a fixed number of time steps ahead.
- Fixed-interval prediction: based on the data from the fixed interval to points in time beyond it.

## 8.6 SMOOTHING

### 8.6.1 Types of Smoothing

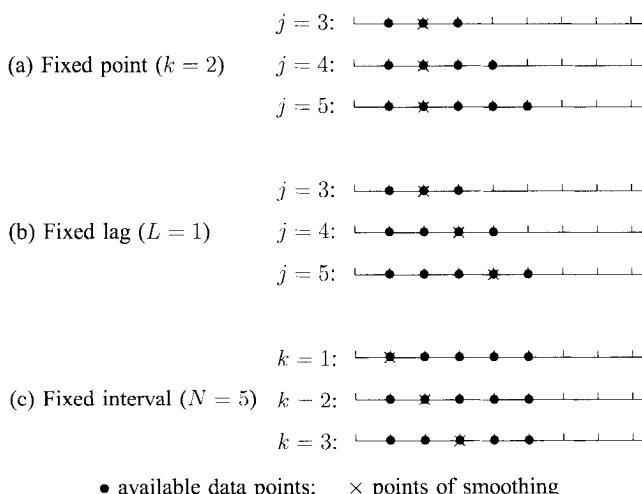
**Smoothing or retrodiction** is the *estimation of the state at time  $k$  within the data interval*, that is, based on data up to  $j > k$ ,

$$\hat{x}(k|j) = E[x(k)|Z^j] \quad (8.6.1-1)$$

There are several types of smoothing:

- (a) **Fixed-point smoothing:**  $k$  is fixed and  $j = k + 1, k + 2, \dots$
- (b) **Fixed-lag smoothing:**  $k$  varies and  $j = k + L$ , where  $L$  is the lag.
- (c) **Fixed-interval smoothing:**  $j = N$  (the data interval is up to  $N$ ) and  $k = 0, 1, \dots, N$ .

These are illustrated in Fig. 8.6.1-1.



**Figure 8.6.1-1:** The three types of smoothing.

The fixed-interval smoothing problem, which is the most common, is discussed in detail next.

## 8.6.2 Fixed-Interval Smoothing

In the following, the *fixed-interval smoothing* algorithm will be derived. This is the most commonly encountered situation where smoothing is of interest — for instance, the smoothing of an entire trajectory based on all the available data after an experiment.

The approach consists of the following:

1. Setting up the posterior pdf of the states at two consecutive times, conditioned on the measurements in the given interval *under the Gaussian assumption*.
2. Maximizing it to yield the MAP estimates of the states, which are then the sought-after conditional means — the smoothed states.

This will yield a *backward recursion* that calculates the smoothed state at time  $k$  from the smoothed state at time  $k + 1$ .

By our definition

$$\hat{x}(k|N) = \arg \max_{x(k)} p[x(k)|Z^N] = \arg \max_{x(k)} p[x(k), Z^N] \quad (8.6.2-1)$$

Consider the joint pdf

$$\begin{aligned} p[x(k), x(k+1), Z^N] &= p[x(k), x(k+1), Z_{k+1}^N, Z^k] \\ &= p[x(k), x(k+1), Z_{k+1}^N | Z^k] p(Z^k) \end{aligned} \quad (8.6.2-2)$$

where

$$Z_i^j \triangleq \{z(l)\}_{l=i}^j \quad (8.6.2-3)$$

The maximization of (8.6.2-2) with respect to  $x(k)$  and  $x(k+1)$  will yield the corresponding smoothed states as defined in (8.6.2-1).

Assume that  $\hat{x}(k|k)$ ,  $k = 1, \dots, N$ , are available from a (“forward”) filtering process that has already been carried out. A backward smoothing recursion will be derived: Assuming that  $\hat{x}(k+1|N)$  is available,  $\hat{x}(k|N)$  will be obtained.

From the above discussion it follows that

$$\hat{x}(k|N) = \arg \max_{x(k)} \left\{ p[x(k), x(k+1), Z^N] \Big|_{x(k+1)=\hat{x}(k+1|N)} \right\} \quad (8.6.2-4)$$

The first term on the right-hand side of (8.6.2-2) can be rewritten as

$$\begin{aligned} p[x(k), x(k+1), Z_{k+1}^N | Z^k] &= p[Z_{k+1}^N | x(k+1), x(k), Z^k] \\ &\quad \cdot p[x(k+1), x(k) | Z^k] \\ &= p[Z_{k+1}^N | x(k+1)] p[x(k+1) | x(k), Z^k] \\ &\quad \cdot p[x(k) | Z^k] \\ &= p[Z_{k+1}^N | x(k+1)] p[x(k+1) | x(k)] \\ &\quad \cdot p[x(k) | Z^k] \end{aligned} \quad (8.6.2-5)$$

The conditioning term dropped in the last line above is irrelevant.

In view of (8.6.2-5), (8.6.2-4) can be written as

$$\hat{x}(k|N) = \arg \max_{x(k)} \left\{ p[x(k+1)|x(k)]p[x(k)|Z^k] \Big|_{x(k+1)=\hat{x}(k+1|N)} \right\} \quad (8.6.2-6)$$

because only the last two terms of (8.6.2-5) contain  $x(k)$ .

Using now the Gaussian assumption, one has

$$p[x(k+1)|x(k)] = \mathcal{N}[x(k+1); F(k)x(k), Q(k)] \quad (8.6.2-7)$$

where  $F(k)$  is the system transition matrix and  $Q(k)$  the process noise covariance,

$$p[x(k)|Z^k] = \mathcal{N}[x(k); \hat{x}(k|k), P(k|k)] \quad (8.6.2-8)$$

where  $P(k|k)$  is the covariance associated with the estimate  $\hat{x}(k|k)$ .

Using the explicit forms of (8.6.2-7) and (8.6.2-8) in (8.6.2-6), and noting that maximization is equivalent to minimization of the negative of the logarithm, (8.6.2-6) can be replaced by

$$\hat{x}(k|N) = \arg \min_{x(k)} J \quad (8.6.2-9)$$

where

$$\begin{aligned} J \triangleq & [\hat{x}(k+1|N) - F(k)x(k)]'Q(k)^{-1}[\hat{x}(k+1|N) - F(k)x(k)] \\ & + [x(k) - \hat{x}(k|k)]'P(k|k)^{-1}[x(k) - \hat{x}(k|k)] \end{aligned} \quad (8.6.2-10)$$

Setting the gradient of  $J$  with respect to  $x(k)$  to zero

$$\begin{aligned} \nabla_{x(k)} J = & -2F(k)'Q(k)^{-1}[\hat{x}(k+1|N) - F(k)x(k)] + 2P(k|k)^{-1} \\ & \cdot [x(k) - \hat{x}(k|k)] \\ = & 0 \end{aligned} \quad (8.6.2-11)$$

yields the smoothed value  $\hat{x}(k|N)$  of  $x(k)$  as

$$[F(k)'Q(k)^{-1}F(k) + P(k|k)^{-1}]\hat{x}(k|N) = F(k)'Q(k)^{-1}\hat{x}(k+1|N) + P(k|k)^{-1}\hat{x}(k|k) \quad (8.6.2-12)$$

Using the matrix inversion lemma

$$(F'Q^{-1}F + P^{-1})^{-1} = P - PF'(FPF' + Q)^{-1}FP \quad (8.6.2-13)$$

in (8.6.2-12) yields, with the arguments of  $F$ ,  $P$  and  $Q$  omitted,

$$\begin{aligned} \hat{x}(k|N) = & [P - PF'(FPF' + Q)^{-1}FP][F'Q^{-1}\hat{x}(k+1|N) + P^{-1}\hat{x}(k|k)] \\ = & [PF'Q^{-1} - PF'(FPF' + Q)^{-1}(FPF' + Q - Q)Q^{-1}]\hat{x}(k+1|N) \\ & + \hat{x}(k|k) - PF'(FPF' + Q)^{-1}F\hat{x}(k|k) \\ = & [PF'Q^{-1} - PF'Q^{-1} + PF'(FPF' + Q)^{-1}]\hat{x}(k+1|N) \\ & + \hat{x}(k|k) - PF'(FPF' + Q)^{-1}F\hat{x}(k|k) \\ = & \hat{x}(k|k) + PF'(FPF' + Q)^{-1}[\hat{x}(k+1|N) - F\hat{x}(k|k)] \end{aligned} \quad (8.6.2-14)$$

From (8.6.2-14), the sought-after backward recursion for the smoothed value of the state — the ***smoothed state*** — is

$$\begin{aligned}\hat{x}(k|N) &= \hat{x}(k|k) + C(k)[\hat{x}(k+1|N) - F(k)\hat{x}(k|k)] \\ k &= N-1, \dots, 0\end{aligned}\quad (8.6.2-15)$$

or

$$\boxed{\hat{x}(k|N) = \hat{x}(k|k) + C(k)[\hat{x}(k+1|N) - \hat{x}(k+1|k)]} \quad (8.6.2-16)$$

where the ***smoother gain*** is

$$C(k) = P(k|k)F(k)'[F(k)P(k|k)F(k)' + Q(k)]^{-1} \quad (8.6.2-17)$$

or

$$\boxed{C(k) = P(k|k)F(k)'P(k+1|k)^{-1}} \quad (8.6.2-18)$$

The initial index for (8.6.2-16) is  $k = N-1$  and  $\hat{x}(k|N)$  follows from  $\hat{x}(k+1|N)$ . This backward recursion starts from the last estimated state  $\hat{x}(N|N)$ . It can be easily shown that (8.6.2-16) holds in exactly the same form even if there is a known input.

The corresponding recursion for the ***smoothed covariance*** (the covariance of the smoothed state)

$$P(k|N) \triangleq E[[x(k) - \hat{x}(k|N)][x(k) - \hat{x}(k|N)]'] \quad (8.6.2-19)$$

is obtained next.

Equation (8.6.2-16) can be rewritten as

$$\hat{x}(k|N) - C(k)\hat{x}(k+1|N) = \hat{x}(k|k) - C(k)F(k)\hat{x}(k|k) \quad (8.6.2-20)$$

Subtracting the above from the identity

$$\begin{aligned}x(k) &= \hat{x}(k|N) + \tilde{x}(k|N) \\ &= \hat{x}(k|j) + \tilde{x}(k|j)\end{aligned}\quad (8.6.2-21)$$

yields

$$\tilde{x}(k|N) + C(k)\hat{x}(k+1|N) = \tilde{x}(k|k) + C(k)F(k)\hat{x}(k|k) \quad (8.6.2-22)$$

The orthogonality properties

$$\tilde{x}(k|N) \perp \hat{x}(k+1|N) \quad (8.6.2-23)$$

$$\tilde{x}(k|k) \perp \hat{x}(k|k) \quad (8.6.2-24)$$

and a vector version of the fact that the mean of the square of a random variable is equal to the square of its mean plus its variance will be used. The (rearranged) vector versions to be used are

$$\begin{aligned}E[\hat{x}(k+1|N)\hat{x}(k+1|N)'] &= E[x(k+1)x(k+1)'] - P(k+1|N) \\ &= F(k)E[x(k)x(k)']F(k)' + Q(k) - P(k+1|N)\end{aligned}\quad (8.6.2-25)$$

and

$$E[\hat{x}(k|k)\hat{x}(k|k)'] = E[x(k)x(k)'] - P(k|k) \quad (8.6.2-26)$$

Then the expected value of the outer product of each side of (8.6.2-22) with itself yields (with all the cross-terms zero after applying the expectation)

$$\begin{aligned} P(k|N) &+ C(k)\{F(k)E[x(k)x(k)']F(k)' + Q(k) - P(k+1|N)\}C(k)' \\ &= P(k|k) + C(k)F(k)\{E[x(k)x(k)'] - P(k|k)\}F(k)'C(k)' \end{aligned} \quad (8.6.2-27)$$

After cancellations and rearrangements in (8.6.2-27), one obtains

$$P(k|N) = P(k|k) + C(k)[P(k+1|N) - F(k)P(k|k)F(k)' - Q(k)]C(k)' \quad (8.6.2-28)$$

or

$$\boxed{P(k|N) = P(k|k) + C(k)[P(k+1|N) - P(k+1|k)]C(k)' \quad k = N-1, \dots, 0} \quad (8.6.2-29)$$

which is the backward recursion for the **smoothed state covariance**. It can be easily shown that the term in the brackets in (8.6.2-29) is negative semidefinite.

Equations (8.6.2-16) and (8.6.2-29) are the backward smoothing recursions.

### Remark

The quantity multiplied by the smoothing gain in (8.6.2-16)

$$\mu(k|N) \triangleq \hat{x}(k+1|N) - \hat{x}(k+1|k) \quad (8.6.2-30)$$

can be interpreted as the **smoothing residual**.

### 8.6.3 Overview of Smoothing

Smoothing requires a backward iteration after the (forward) filtering has been performed and its results

$$\hat{x}(k|k), \quad \hat{x}(k+1|k), \quad P(k|k), \quad P(k+1|k) \quad k = 0, 1, \dots, N-1 \quad (8.6.3-1)$$

have been stored.

The iteration consists of the smoother gain calculation

$$\boxed{C(k) = P(k|k)F(k)'P(k+1|k)^{-1}} \quad (8.6.3-2)$$

followed by the evaluation of the smoothed state

$$\boxed{\hat{x}(k|N) = \hat{x}(k|k) + C(k)[\hat{x}(k+1|N) - \hat{x}(k+1|k)]} \quad (8.6.3-3)$$

and the covariance of the smoothed state

$$\boxed{P(k|N) = P(k|k) + C(k)[P(k+1|N) - P(k+1|k)]C(k)' \quad k = N-1, \dots, 0} \quad (8.6.3-4)$$

for  $k = N-1, \dots, 1, 0$ .

## 8.6.4 Smoothing — Summary

Smoothing (or retrodiction) is the estimation of the state at a time within the data interval.

There are several types of smoothing:

- Fixed-point smoothing: The time of interest is fixed and the data interval varies.
- Fixed-lag smoothing: The time of interest is  $L$  steps (the lag) behind the end of the data interval, which varies.
- Fixed-interval smoothing: The time of interest varies within the data interval.

The most relevant algorithm — for fixed-interval smoothing — has been presented in full.

## 8.7 NOTES AND PROBLEMS

### 8.7.1 Bibliographical Notes

The use of autocorrelated noise to model maneuvers, presented in Section 8.2, is based on [Singer70]. The steady-state solution for the filter gains in explicit form, similar to the alpha-beta-gamma filter, is available in [Fitzgerald81].

The situation of cross-correlated noise sequences and colored measurement noise has been treated in [Bryson75, Sage71, Anderson79, Maybeck82]. These references also discuss prediction and smoothing. The fixed-interval smoothing problem discussed in Section 8.6 follows the development of [Rauch65]. As shown in [Fraser69], the smoothed state can also be computed as a linear combination of the estimate computed from (standard) forward Kalman filter and an estimate computed from a backward running Kalman filter.

### 8.7.2 Problems

#### 8-1 Fixed-point prediction — the mean and covariance.

1. Prove (8.5.2-4).
2. Prove (8.5.2-5). *Hint:* Add and subtract  $x(N)$  and use the fact that

$$\hat{x}(N|k) \perp \tilde{x}(N|i) \quad \forall i > k$$

#### 8-2 Updated state covariance conditioned on prior data.

Find, for  $k < N$ ,  $\text{cov}[\hat{x}(N|N)|Z^k]$ .

#### 8-3 Acceleration variance in the mixed pmf/pdf model.

Show how (8.2.2-3) follows from (8.2.2-2).

#### 8-4 Limiting process of an autocorrelated noise to white noise.

Prove (8.2.2-7). *Hint:* Define the limit such that the power spectral density at frequency zero stays the same.

- 8-5 Prewhitenning of a colored noise — the shaping filter.** Given the zero-mean random sequence with autocorrelation function

$$E[w(i)w(j)] = \delta_{ij} + \alpha(\delta_{i+1,j} + \delta_{i-1,j})$$

find its prewhitening system, that is, the causal linear system driven by white noise whose output has this autocorrelation.

- 8-6 Smoothing compared with estimation.**

1. For problem 5-1(4), with the same noise sequence as used in the forward estimation, calculate and list

$$\hat{x}(k|50), \tilde{x}(k|50), P(k|50), \tilde{x}(k|50)/\sqrt{P(k|50)}, \quad k = 1, \dots, 50$$

2. Compare  $P(k|k)$  with  $P(k|50)$ .

- 8-7 Dyad of a state estimate — the expected value.** Prove the identities (8.6.2-25) and (8.6.2-26).

- 8-8 Mixed pmf/pdf of the acceleration model.** Derive (8.2.2-2) from the three assumptions right above it.

- 8-9 Correlation between current estimate and smoothed value.** Find the expression of the correlation  $E[\hat{x}(k|k+1)\hat{x}(k+1|k+1)']$ .

- 8-10 Smoothing compared with estimation — a simple case.**

Given the scalar initial state

$$x(0) \sim \mathcal{N}[\hat{x}(0|0), P(0|0)]$$

the system equation

$$x(k+1) = x(k) + v(k) \quad k = 1, \dots$$

and the measurement equation

$$z(k) = x(k) + w(k) \quad k = 1, \dots$$

with both noises zero-mean, unity variance, white and mutually uncorrelated and uncorrelated with the initial state.

Assuming  $P(0|0) = 2$ , calculate

1.  $P(1|1)$
2.  $P(1|2)$ .

- 8-11 Prewhitenning of a second-order process.** Given the stationary stochastic process with power spectrum  $\frac{\alpha^2}{\omega^4 + \omega_j^4}$ , find its prewhitening system or shaping filter — the LTI system driven by white noise, whose output has this spectrum. Hint: Use a second-order transfer function with coefficients to be determined.



# Chapter 9

## CONTINUOUS-TIME LINEAR STATE ESTIMATION

### 9.1 INTRODUCTION

#### 9.1.1 Outline

This chapter deals with the estimation of the state of continuous-time linear dynamic systems based on observations which are made also in continuous time.

The linear minimum mean square error (LMMSE) filter for this *continuous-time* problem, known as the *Kalman-Bucy filter*, is obtained in Section 9.2 from a limiting process of the discrete time problem. The properties of the innovation process are derived. The asymptotic properties of the estimated state covariance equation — the continuous-time Riccati equation — which yields a steady-state filter, are also presented.

Section 9.3 deals with the prediction of the state of a continuous-time system.

The *duality*<sup>1</sup> of the LMMSE estimation with the linear-quadratic (LQ) control problem is discussed in Section 9.4. These two problems have their solutions determined by the same Riccati equation.

The Wiener-Hopf problem, which consists of the estimation of a stochastic process based on another observed process using their auto- and cross-covariances, is the topic of a brief discussion in Section 9.5.

#### 9.1.2 Continuous-Time Estimation — Summary of Objectives

- Derive the LMMSE filter for continuous time dynamic systems — the Kalman-Bucy filter — from a limiting process of the discrete-time filter.

---

<sup>1</sup>This duality is conceptually similar to the one discussed in Section 7.2.

- Present the asymptotic properties of the continuous-time Riccati matrix differential equation and the convergence of the filter to steady state.
- Discuss the state prediction.
- Show the duality between the following problems:
  - LMMSE estimation
  - LQ control.
- Briefly review the Wiener-Hopf problem — LMMSE estimation for stationary stochastic processes — and indicate its relationship with the Kalman-Bucy filter.

## 9.2 THE CONTINUOUS-TIME LINEAR STATE ESTIMATION FILTER

### 9.2.1 The Continuous-Time Estimation Problem

The state equation is (without input, for simplicity)

$$\dot{x}(t) = A(t)x(t) + D(t)\tilde{v}(t) \quad (9.2.1-1)$$

with  $\tilde{v}(t)$  the process noise, zero mean and white

$$E[\tilde{v}(t)] = 0 \quad (9.2.1-2)$$

$$E[\tilde{v}(t)\tilde{v}(\tau)'] = \tilde{Q}(t)\delta(t - \tau) \quad (9.2.1-3)$$

where  $\tilde{Q}(t)$  is the (possibly time-varying) **intensity of the white noise**. This becomes the power spectral density if it is time-invariant.

The measurement equation is

$$z(t) = C(t)x(t) + \tilde{w}(t) \quad (9.2.1-4)$$

with the process noise  $\tilde{w}(t)$  zero mean and white

$$E[\tilde{w}(t)] = 0 \quad (9.2.1-5)$$

$$E[\tilde{w}(t)\tilde{w}(\tau)'] = \tilde{R}(t)\delta(t - \tau) \quad (9.2.1-6)$$

The process noise, the measurement noise and the initial state are mutually uncorrelated. Note that the noises can be *nonstationary*.

The continuous time filter will be derived by taking the limit of the discrete-time filter as  $\Delta \triangleq (t_{k+1} - t_k) \rightarrow 0$ .

## Ito Differential Equation Form

The state equation is sometimes written in **Ito differential equation** form

$$dx(t) = A(t)x(t) dt + D(t) d\mathbf{w}_v(t) \quad (9.2.1-7)$$

where  $d\mathbf{w}_v(t)$  is the infinitesimal increment of the Wiener process  $\mathbf{w}_v(t)$  with moments

$$E[d\mathbf{w}_v(t)] = 0 \quad (9.2.1-8)$$

and

$$E[d\mathbf{w}_v(t)d\mathbf{w}_v(t)'] = \tilde{Q}(t) dt \quad (9.2.1-9)$$

The incremental Wiener process in (9.2.1-7) is

$$d\mathbf{w}_v(t) = \tilde{v}(t) dt \quad (9.2.1-10)$$

Equation (9.2.1-7) is more rigorous than (9.2.1-1), since the white noise exists only in the sense of generalized functions (the Wiener process is nondifferentiable).

Similarly, the measurement equation (9.2.1-4) can be written as

$$d\zeta(t) \triangleq z(t)dt = C(t)x(t) dt + d\mathbf{w}_w(t) \quad (9.2.1-11)$$

where

$$d\mathbf{w}_w(t) = \tilde{w}(t) dt \quad (9.2.1-12)$$

and

$$E[d\mathbf{w}_w(t)d\mathbf{w}_w(t)'] = \tilde{R}(t) dt \quad (9.2.1-13)$$

## 9.2.2 Connection Between Continuous - and Discrete-Time Representations and Their Noise Statistics

The discrete-time state equation is, with  $F(\cdot)$  denoting the transition matrix,

$$x(t_{k+1}) = F(t_{k+1}, t_k)x(t_k) + v(t_k) \quad (9.2.2-1)$$

The connection between the continuous- and discrete-time representations is, taking a small time increment  $\Delta = t_{k+1} - t_k$ , given by

$$\begin{aligned} \dot{x}(t_k) &= \frac{1}{\Delta}[x(t_k + \Delta) - x(t_k)] = \frac{1}{\Delta}[F(t_k + \Delta, t_k)x(t_k) + v(t_k) - x(t_k)] \\ &= \frac{1}{\Delta}[F(t_k + \Delta, t_k) - I]x(t_k) + \frac{1}{\Delta}v(t_k) \end{aligned} \quad (9.2.2-2)$$

Using the following two properties of the transition matrix

$$F(t, t) = I \quad (9.2.2-3)$$

$$\frac{\partial}{\partial \tau} F(\tau, t) = A(\tau) F(\tau, t) \quad (9.2.2-4)$$

one has, as  $\Delta \rightarrow 0$ ,

$$\frac{1}{\Delta} [F(t + \Delta, t) - I] = A(t) \quad (9.2.2-5)$$

The expression of the discrete-time process noise in (9.2.2-1) is

$$v(t_k) = \int_{t_k}^{t_{k+1}} F(t_{k+1}, \tau) D(\tau) \tilde{v}(\tau) d\tau \quad (9.2.2-6)$$

Using the fact that, according to (9.2.2-3), the transition matrix over a very short interval tends to the identity matrix, one has from (9.2.2-6)

$$D(t_k) \tilde{v}(t_k) = \frac{1}{\Delta} v(t_k) \quad (9.2.2-7)$$

which then completes the proof of equivalence between (9.2.2-2) and (9.2.1-1).

The covariance of the discrete-time process noise is

$$Q(t_k) = E[v(t_k)v(t_k)'] = E \left[ \left[ \int_{t_k}^{t_k + \Delta} D(\tau_1) \tilde{v}(\tau_1) d\tau_1 \right] \left[ \int_{t_k}^{t_k + \Delta} D(\tau_2) \tilde{v}(\tau_2) d\tau_2 \right]' \right] \quad (9.2.2-8)$$

Rewriting the above as a double integral (with the limits omitted) and applying the expectation on the integrand yields, using (9.2.1-3),

$$\begin{aligned} Q(t_k) &= \int \int D(\tau_1) \tilde{Q}(\tau_1) D(\tau_2)' \delta(\tau_1 - \tau_2) d\tau_1 d\tau_2 = \int D(\tau_1) \tilde{Q}(\tau_1) D(\tau_1)' d\tau_1 \\ &= D(t_k) \tilde{Q}(t_k) D(t_k)' \Delta \end{aligned} \quad (9.2.2-9)$$

and thus

$$D(t_k) \tilde{Q}(t_k) D(t_k)' = \frac{1}{\Delta} Q(t_k) \quad (9.2.2-10)$$

### Remark

It follows from (9.2.1-3) that the **physical dimension** of the continuous-time process noise intensity  $\tilde{Q}$  is the physical dimension of  $\tilde{v}$  squared multiplied by time, since the physical dimension of  $\delta(t)$  is  $t^{-1}$ . This is consistent with  $\tilde{Q}$  being a power spectral density with the physical dimension of  $\tilde{v}$  squared divided by frequency.

On the other hand, the physical dimension of the discrete-time intensity (covariance)  $Q$  is the physical dimension of  $v$  squared. This is the reason for the similarity of (9.2.2-10) and (9.2.2-7): Both the discrete-time noise and its covariance are divided by the time interval  $\Delta$  to yield their continuous-time counterparts.

The **discrete-time measurement** can be viewed as a **short-term average** of the continuous-time measurement

$$z(t_k) = \frac{1}{\Delta} \int_{t_k}^{t_k + \Delta} z(\tau) d\tau = C(t_k)x(t_k) + \frac{1}{\Delta} \int_{t_k}^{t_k + \Delta} \tilde{w}(\tau) d\tau \quad (9.2.2-11)$$

Comparing the above with the discrete time measurement equation

$$z(t_k) = H(t_k)x(t_k) + w(t_k) \quad (9.2.2-12)$$

yields

$$H(t_k) = C(t_k) \quad (9.2.2-13)$$

and the relationship between the discrete time and continuous time measurement noises is

$$w(t_k) = \frac{1}{\Delta} \int_{t_k}^{t_k + \Delta} \tilde{w}(\tau) d\tau \quad (9.2.2-14)$$

The relationship between the covariance of the discrete time measurement noise and the intensity of the continuous time measurement noise is obtained, similarly to (9.2.2-8), as

$$\begin{aligned} R(t_k) &= E[w(t_k)w(t_k)'] = E\left[\frac{1}{\Delta^2} \int \int \tilde{w}(\tau_1)\tilde{w}(\tau_2)' d\tau_1 d\tau_2\right] \\ &= \frac{1}{\Delta^2} \int \int \tilde{R}(\tau_1)\delta(\tau_1 - \tau_2) d\tau_1 d\tau_2 \\ &= \frac{1}{\Delta^2} \int \tilde{R}(\tau_1) d\tau_1 \\ &= \frac{1}{\Delta} \tilde{R}(t_k) \end{aligned} \quad (9.2.2-15)$$

or

$$\tilde{R}(t_k) = R(t_k)\Delta \quad (9.2.2-16)$$

### 9.2.3 The Continuous-Time Filter Equations

Differencing the discrete-time filter (5.2.3-12), one has

$$\begin{aligned} \hat{x}(k+1|k+1) - \hat{x}(k|k) &= F(k)\hat{x}(k|k) - \hat{x}(k|k) \\ &\quad + W(k+1)[z(k+1) - H(k+1)\hat{x}(k+1|k)] \end{aligned} \quad (9.2.3-1)$$

Replacing  $F(k)$  by  $F(t_{k+1}, t_k)$  with the explicit full notation, using expression (5.2.3-17) for the filter gain and identity (9.2.2-13), yields, after division by  $\Delta$ ,

$$\begin{aligned} \frac{1}{\Delta} [\hat{x}(t_{k+1}|t_{k+1}) - \hat{x}(t_k|t_k)] &= \frac{1}{\Delta} [F(t_{k+1}, t_k) - I]\hat{x}(t_k|t_k) \\ &\quad + P(t_{k+1}|t_{k+1})C(t_{k+1})'[R(t_{k+1})\Delta]^{-1} \\ &\quad \cdot [z(t_{k+1}) - C(t_k)\hat{x}(t_{k+1}|t_k)] \end{aligned} \quad (9.2.3-2)$$

As  $\Delta \rightarrow 0$ , using (9.2.2-5) and (9.2.2-15) and the fact that

$$\hat{x}(t_{k+1}|t_k) - \hat{x}(t_k|t_k) = O(\Delta) \quad (9.2.3-3)$$

that is, this difference is of the order of  $\Delta$  and

$$\hat{x}(t_{k+1}|t_k) \xrightarrow{\Delta \rightarrow 0} \hat{x}(t_k|t_k) \triangleq \hat{x}(t_k) \quad (9.2.3-4)$$

one obtains, after dropping the subscript of  $t$ ,

$$\boxed{\dot{\hat{x}}(t) = A(t)\hat{x}(t) + L(t)[z(t) - C(t)\hat{x}(t)]} \quad (9.2.3-5)$$

where the (optimal) *continuous-time filter gain* is

$$L(t) \triangleq P(t)C(t)'R(t)^{-1} \quad (9.2.3-6)$$

This is the *differential equation* of the *continuous-time state estimate*, namely, the *continuous time state estimation filter*, called the **Kalman-Bucy filter**.

The discrete-time covariance update equation (5.2.3-14) can be written using (5.2.3-17) as

$$\begin{aligned} P(t_{k+1}|t_{k+1}) &= P(t_{k+1}|t_k) - P(t_{k+1}|t_{k+1})H(t_{k+1})'R(t_{k+1})^{-1}H(t_{k+1})P(t_{k+1}|t_k) \\ &= F(t_{k+1}, t_k)P(t_k|t_k)F(t_{k+1}, t_k)' + Q(t_k) \\ &\quad - P(t_{k+1}|t_{k+1})H(t_{k+1})'R(t_{k+1})^{-1}H(t_{k+1})P(t_{k+1}|t_k) \\ &= [I + A(t_k)\Delta]P(t_k|t_k)[I + A(t_k)\Delta]' + Q(t_k) \\ &\quad - P(t_{k+1}|t_{k+1})H(t_{k+1})'R(t_{k+1})^{-1}H(t_{k+1})P(t_{k+1}|t_k) \\ &= P(t_k|t_k) + A(t_k)P(t_k|t_k)\Delta + P(t_k|t_k)A(t_k)'\Delta + Q(t_k) \\ &\quad - P(t_{k+1}|t_{k+1})H(t_{k+1})'R(t_{k+1})^{-1}H(t_{k+1})P(t_{k+1}|t_k) \end{aligned} \quad (9.2.3-7)$$

where the term in  $\Delta^2$  has been neglected. In difference form, one has

$$\begin{aligned} \frac{1}{\Delta}[P(t_{k+1}|t_{k+1}) - P(t_k|t_k)] &= A(t_k)P(t_k|t_k) + P(t_k|t_k)A(t_k)' + \frac{1}{\Delta}Q(t_k) \\ &\quad - P(t_{k+1}|t_{k+1})H(t_{k+1})'[R(t_{k+1})\Delta]^{-1} \\ &\quad \cdot H(t_{k+1})P(t_{k+1}|t_k) \end{aligned} \quad (9.2.3-8)$$

Using (9.2.2-9), (9.2.2-12) and (9.2.2-15) and the fact that, under the assumption of continuity,

$$P(t_{k+1}|t_k) \xrightarrow{\Delta \rightarrow 0} P(t_{k+1}|t_{k+1}) \triangleq P(t_{k+1}) \quad (9.2.3-9)$$

yields, after dropping the subscript of  $t$ ,

$$\boxed{\dot{P}(t) = A(t)P(t) + P(t)A(t)' + D(t)\tilde{Q}(t)D(t)' - P(t)C(t)'R(t)^{-1}C(t)P(t)} \quad (9.2.3-10)$$

The above is the **matrix Riccati differential equation** for the state estimate covariance. Its solution can be obtained using numerical techniques.

## Remarks

The structure of the Kalman-Bucy filter, as given in (9.2.3-5), is the same as that of the Luenberger observer [Chen84].

Note that there is no prediction stage — this is a direct consequence of the continuous-time formulation.

The measurement noise covariance  $\tilde{R}$  has to be invertible — no “perfect” measurements are allowed in the Kalman-Bucy filter. If some of the components of the measurement are noiseless (the measurement noise covariance has some zero eigenvalues), then one can use a so-called observer-estimator or assume a “small” noise to be present in those components. In this case, one can “isolate” some state components for which there is no need to carry out estimation.

In the absence of measurements, which can be modeled by letting  $\tilde{R}^{-1} = 0$ , the state estimate and its covariance evolve according to the following “open-loop” equations

$$\dot{\hat{x}}(t) = A(t)\hat{x}(t) \quad (9.2.3-11)$$

$$P(t) = A(t)P(t) + P(t)A(t)' + D(t)\tilde{Q}(t)D(t)' \quad (9.2.3-12)$$

obtained from (9.2.3-5) and (9.2.3-10), respectively, by dropping the last term in each. Equation (9.2.3-12) is known as the *Lyapunov differential equation*.

The above two equations are used later in the *continuous-discrete filter*, for the case of continuous-time state equation and discrete-time measurements.

The *Ito differential equation* form of the filter (9.2.3-5) is

$$\begin{aligned} d\hat{x}(t) &= A(t)\hat{x}(t)dt + L(t)[z(t) - C(t)\hat{x}(t)]dt \\ &= [A(t) - L(t)C(t)]\hat{x}(t)dt + L(t)d\zeta(t) \end{aligned} \quad (9.2.3-13)$$

where the last form uses the notation (9.2.1-11).

## 9.2.4 The Continuous-Time Innovation

The *continuous-time innovation* is, from (9.2.3-4),

$$\nu(t) \triangleq z(t) - C(t)\hat{x}(t) = C(t)x(t) + \tilde{w}(t) - C(t)\hat{x}(t) = C(t)\tilde{x}(t) + \tilde{w}(t) \quad (9.2.4-1)$$

where

$$\tilde{x}(t) \triangleq x(t) - \hat{x}(t) \quad (9.2.4-2)$$

is the *state estimation error*.

The fact that the continuous-time innovation is *zero mean* follows immediately from the limiting process used to derive the continuous-time filter.

The continuous-time innovation is also *white* as it is shown next. The autocorrelation of the innovation is

$$\begin{aligned} E[\nu(t)\nu(\tau)'] &= E[[C(t)\tilde{x}(t) + \tilde{w}(t)][C(\tau)\tilde{x}(\tau) + \tilde{w}(\tau)]]' \\ &= E[\tilde{x}(\tau)\tilde{w}(t)']' C(\tau)' + C(t)E[\tilde{x}(t)\tilde{w}(\tau)'] \\ &\quad + C(t)E[\tilde{x}(t)\tilde{x}(\tau)'] C(\tau)' + E[\tilde{w}(t)\tilde{w}(\tau)'] \end{aligned} \quad (9.2.4-3)$$

Assume  $t > \tau$ . The first term after the last equal sign above is zero since  $\tilde{w}(t)$  is white and zero mean, and thus orthogonal to  $\tilde{x}(\tau)$ ,  $\forall \tau < t$ . Also note that

$$E[\tilde{x}(t)\tilde{x}(\tau)'] = E[\tilde{x}(t)[x(\tau) - \hat{x}(\tau)]]' = E[\tilde{x}(t)x(\tau)'] \quad (9.2.4-4)$$

since  $\tilde{x}(t)$  is orthogonal to any function of the measurements up to  $\tau$ ,  $\forall \tau \leq t$ , and thus on  $\hat{x}(\tau)$ .

Using these results and the fact that

$$E[\tilde{w}(t)\tilde{w}(\tau)'] = \tilde{R}(t)\delta(t - \tau) \quad (9.2.4-5)$$

in (9.2.4-3) yields

$$\begin{aligned} E[\nu(t)\nu(\tau)'] &= C(t)E[\tilde{x}(t)\tilde{w}(\tau)'] + C(t)E[\tilde{x}(t)x(\tau)']C(\tau)' + \tilde{R}(t)\delta(t - \tau) \\ &= C(t)E[\tilde{x}(t)[C(\tau)x(\tau) + \tilde{w}(\tau)]]' + \tilde{R}(t)\delta(t - \tau) \\ &= \tilde{R}(t)\delta(t - \tau) + C(t)E[\tilde{x}(t)z(\tau)'] \end{aligned} \quad (9.2.4-6)$$

Again, since

$$\tilde{x}(t) \perp z(\tau) \quad \forall \tau \leq t \quad (9.2.4-7)$$

the last term in (9.2.4-6) vanishes.

Thus the autocorrelation of the continuous-time innovation is

$E[\nu(t)\nu(\tau)'] = \tilde{R}(t)\delta(t - \tau)$

(9.2.4-8)

that is, the innovation is white. Furthermore, the intensity of the continuous-time innovation is *equal* to the intensity of the measurement noise.

### Remark

Note the difference from the discrete-time case where the innovation covariance is

$$S(k) = E[\nu(k)\nu(k)'] = H(k)P(k|k-1)H(k)' + R(k) \quad (9.2.4-9)$$

that is, *larger* than the measurement noise covariance  $R(k)$ , in the sense that the difference between them is positive semidefinite. This difference is due to the fact that in discrete time there is a predicted state distinct from the (current or updated) state estimate. In contrast, in continuous time there is no prediction since the measurements arrive continuously.

## 9.2.5 Asymptotic Properties of the Continuous-Time Riccati Equation

### Time-Varying Systems

The continuous-time **matrix Riccati differential equation** (9.2.3-9) is, for a time-varying system,

$$\dot{P}(t) = A(t)P(t) + P(t)A(t)' + D(t)\tilde{Q}(t)D(t)' - P(t)C(t)'\tilde{R}(t)^{-1}C(t)P(t) \quad (9.2.5-1)$$

Consider (9.2.5-1) for a system *without process noise*. Then, using the identity

$$\frac{d}{dt}[P(t)^{-1}] = -P(t)^{-1}\dot{P}(t)P(t)^{-1} \quad (9.2.5-2)$$

one can rewrite (9.2.5-1), with  $\tilde{Q} = 0$  and without the time arguments for simplicity, as (see problem 9-1)

$$\frac{d}{dt}(P^{-1}) = -A'P^{-1} - P^{-1}A + C'\tilde{R}^{-1}C \quad (9.2.5-3)$$

The above equation for the inverse covariance or **information matrix** has the following explicit solution

$$P(t)^{-1} = F(t_0, t)'P(t_0)^{-1}F(t_0, t) + \int_{t_0}^t F(\tau, t)'C(\tau)'\tilde{R}(\tau)^{-1}C(\tau)F(\tau, t)d\tau \quad (9.2.5-4)$$

where  $F$  is the transition matrix corresponding to  $A(t)$  (see problem 9-2).

Equation (9.2.5-4) indicates that, even if  $P(t_0)^{-1}$  is singular (i.e., no prior information is available about some of the state components), such information will become available from the observations if the last term in (9.2.5-4), called the **observability Gramian**

$$\mathbf{G} \triangleq \int_{t_0}^t F(\tau, t)'C(\tau)'\tilde{R}(\tau)^{-1}C(\tau)F(\tau, t)d\tau \quad (9.2.5-5)$$

is *positive definite*.

It can be observed from (9.2.5-4), which pertains to the situation without process noise, that as  $t \rightarrow \infty$  the information matrix can also tend to infinity, in which case the covariance will tend to zero.

### Time-Invariant Systems

For a *time-invariant system*, the positive definiteness of the Gramian (9.2.5-5) is equivalent to the condition that the **observability matrix** of the pair  $(A, C)$  given by

$$\mathbf{Q}_O \triangleq [C' \ A'C' \ \cdots \ (A')^{n_x-1}C']' \quad (9.2.5-6)$$

where  $n_x$  is the dimension of the state, is of full rank (i.e.,  $n_x$ ).

Observability of the system guarantees the positive definiteness of the inverse covariance matrix, which makes it invertible, that is, it guarantees the existence (boundedness) of the covariance matrix.

The asymptotic properties of the solution of the Riccati equation for a time-invariant system *with process noise* are:

1. Under the observability condition of the pair  $(A, C)$ , the covariance converges to a *finite steady-state covariance*.
2. If, in addition, with  $\tilde{D}$  being a square root of the process noise intensity matrix premultiplied by  $D$  and postmultiplied by  $D'$ , the pair  $(A, \tilde{D})$  is controllable, then the resulting steady-state covariance is *unique and positive definite*.

The matrix square root  $\tilde{D}$  is defined in this case by

$$\tilde{D}\tilde{D}' \triangleq D\tilde{Q}D' \quad (9.2.5-7)$$

The controllability condition for the pair  $(A, \tilde{D})$  is that the matrix

$$\mathcal{Q}_C \triangleq [\tilde{D} \ A\tilde{D} \ \cdots \ A^{n_x-1}\tilde{D}] \quad (9.2.5-8)$$

is of full rank (i.e.,  $n_x$ ).

The steady-state covariance is the solution of the **algebraic matrix Riccati equation**

$$AP + PA' + D\tilde{Q}D' - PC'\tilde{R}^{-1}CP = 0 \quad (9.2.5-9)$$

which, under conditions 1 and 2, has a *unique positive definite solution*. To this corresponds a **steady-state filter**.

## Stability of the Filter

Consider the filter equation

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x}(t) + P(t)C'\tilde{R}^{-1}[z(t) - C\hat{x}(t)] \\ &= A\hat{x}(t) + P(t)C'\tilde{R}^{-1}[Cx(t) + \tilde{w}(t) - C\hat{x}(t)] \end{aligned} \quad (9.2.5-10)$$

and the system equation

$$\dot{x}(t) = Ax(t) + D\tilde{v}(t) \quad (9.2.5-11)$$

Subtracting (9.2.5-10) from (9.2.5-11) and using the notation

$$\tilde{x} \triangleq x - \hat{x} \quad (9.2.5-12)$$

for the state estimation error, it follows that this error obeys the following differential equation

$$\begin{aligned} \dot{\tilde{x}}(t) &= A\tilde{x}(t) - P(t)C'\tilde{R}^{-1}C\tilde{x}(t) - P(t)C'\tilde{R}^{-1}\tilde{w}(t) + D\tilde{v}(t) \\ &= [A - P(t)C'\tilde{R}^{-1}C]\tilde{x}(t) - P(t)C'\tilde{R}^{-1}\tilde{w}(t) + D\tilde{v}(t) \end{aligned} \quad (9.2.5-13)$$

Under conditions 1 and 2 — which guarantee that (9.2.5-13) has a stationary solution (the expected value of  $\tilde{x}\tilde{x}'$ , which is the error covariance, is bounded) — the system matrix for the error equation

$$\tilde{A} \triangleq A - P(t)C'\tilde{R}^{-1}C \quad (9.2.5-14)$$

is *stable* (i.e., all its eigenvalues are in the left half plane).

Note that there is *no stability requirement* for the dynamic system (9.2.5-11).

## 9.2.6 Examples of Continuous-Time Filters

### A scalar system

Consider the scalar system

$$\dot{x} = -ax(t) + \tilde{v}(t) \quad (9.2.6-1)$$

with

$$E[\tilde{v}(t)] = 0 \quad (9.2.6-2)$$

$$E[\tilde{v}(t)\tilde{v}(\tau)] = \tilde{q}\delta(t - \tau) \quad (9.2.6-3)$$

and the measurement

$$z(t) = x(t) + \tilde{w}(t) \quad (9.2.6-4)$$

with

$$E[\tilde{w}(t)] = 0 \quad (9.2.6-5)$$

$$E[\tilde{w}(t)\tilde{w}(\tau)] = \tilde{r}\delta(t - \tau) \quad (9.2.6-6)$$

and the two noises and the initial state uncorrelated from each other.

The Riccati equation for the variance  $p(t)$  of the filtered state is

$$\dot{p} = -2ap + \tilde{q} - \frac{p^2}{\tilde{r}} \quad (9.2.6-7)$$

with initial condition  $p_0$ , the initial state variance.

The scalar Riccati equation (9.2.6-7) has the following explicit solution:

$$p(t) = p_1 + \frac{p_1 + p_2}{\frac{p_0 + p_2}{p_0 - p_1}e^{2\alpha t} - 1} \quad (9.2.6-8)$$

where

$$\alpha = \sqrt{a^2 + \frac{\tilde{q}}{\tilde{r}}} \quad (9.2.6-9)$$

$$p_1 = \tilde{r}(\alpha - a) \quad (9.2.6-10)$$

$$p_2 = \tilde{r}(\alpha + a) \quad (9.2.6-11)$$

The resulting Kalman-Bucy filter is

$$\dot{\hat{x}}(t) = -a\hat{x}(t) + \frac{p(t)}{r}[z(t) - \hat{x}(t)] \quad (9.2.6-12)$$

## The Steady-state filter and its transfer function

Since this is a time-invariant system with stationary noises and the conditions for the existence and positiveness of the steady-state solution are satisfied, the filter itself will become time-invariant after the transient — the **steady-state filter** — regardless of whether the dynamic system is stable.

The solution of the Riccati equation, which is the optimal state estimate variance, has the asymptotic value

$$\lim_{t \rightarrow \infty} p(t) = p_1 \quad (9.2.6-13)$$

It can be easily shown that this value is also the only positive solution of the algebraic Riccati equation

$$0 = -2ap + \tilde{q} - \frac{p^2}{\tilde{r}} \quad (9.2.6-14)$$

This yields the steady-state filter with the fixed gain  $p_1/\tilde{r}$ . The differential equation of the filter is

$$\dot{\hat{x}}(t) = -a\hat{x}(t) + \frac{p_1}{\tilde{r}}[z(t) - \hat{x}(t)] \quad (9.2.6-15)$$

Since this is time-invariant, one can obtain its Laplace transfer function, which is

$$\frac{\hat{X}(s)}{Z(s)} = \frac{\alpha - a}{s + \alpha} \quad (9.2.6-16)$$

which is clearly seen as a stable system, since  $\alpha > 0$ .

## A second-order system

Consider the following system, which is an oscillator (with poles at  $\pm j$ )

$$\dot{x}(t) = Ax(t) + D\tilde{v}(t) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix}\tilde{v}(t) \quad (9.2.6-17)$$

with measurement

$$z(t) = [1 \ 0]x(t) + \tilde{w}(t) = Cx(t) + \tilde{w}(t) \quad (9.2.6-18)$$

with noise power spectral densities  $\tilde{q} = 1$  and  $\tilde{r} = 3$ , respectively.

The Riccati differential equation for the state estimation covariance is

$$\dot{P}(t) = AP(t)P(t)A' - P(t)C'\tilde{r}^{-1}CP(t) + D\tilde{q}D' \quad (9.2.6-19)$$

which becomes in steady-state the algebraic Riccati equation

$$0 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}P + P\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} - \frac{1}{3}P\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}P + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad (9.2.6-20)$$

This leads to the following system of quadratic equations for the components of  $P$

$$\begin{aligned} 2P_{12} - \frac{1}{3}P_{11}^2 &= 0 \\ P_{22} - P_{11} - \frac{1}{3}P_{12}P_{11} &= 0 \\ -2P_{12} - \frac{1}{3}P_{12}^2 &= 1 \end{aligned} \quad (9.2.6-21)$$

which can be easily solved to obtain

$$P_{12} = -3 \pm 2\sqrt{3} \quad (9.2.6-22)$$

from which only the first solution  $P_{12} \approx 0.464$  is viable because otherwise  $P_{11}$  is imaginary. The other solutions subject to positivity are  $P_{11} = 1.67$  and  $P_{22} = 2.59$ . It can be easily shown that the resulting  $P$  is positive definite. The filter gain is then

$$W = PC'\tilde{r}^{-1} = \frac{1}{3} \begin{bmatrix} 1.67 \\ 0.464 \end{bmatrix} = \begin{bmatrix} 0.557 \\ 0.155 \end{bmatrix} \quad (9.2.6-23)$$

Thus the steady state filter equation is

$$\dot{\hat{x}}(t) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \hat{x}(t) + \begin{bmatrix} 0.557 \\ 0.155 \end{bmatrix} [z(t) - \hat{x}_1(t)] \quad (9.2.6-24)$$

The filter transfer function from  $z$  to  $\hat{x}_1$  is

$$\frac{\hat{X}_1(s)}{Z(s)} = \frac{0.557s + 0.155}{s^2 + 0.557s + 1.155} \quad (9.2.6-25)$$

### On the Solution of the Riccati Equation

For dimension greater than one, the Riccati equation cannot be solved, in general, explicitly. There are some techniques that transform the  $n_x$ -dimensional nonlinear matrix Riccati equation into a  $2n_x$ -dimensional linear equation, but even that requires, in general, numerical solution. The most common method of solving the continuous-time Riccati equation is via numerical integration.

The steady-state solution of the Riccati equation can be obtained for low-dimension kinematic models, similarly to the discrete-time case, in closed form.

### 9.2.7 Overview of the Kalman-Bucy Filter

Figure 9.2.7-1 presents the block diagram of the Kalman-Bucy filter.

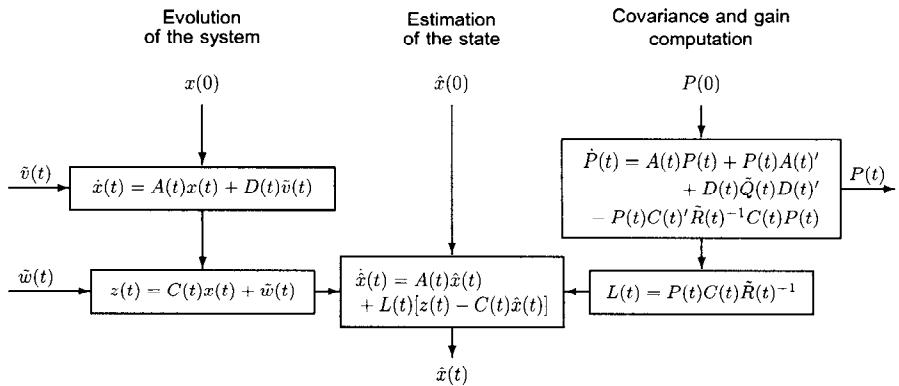


Figure 9.2.7-1: Block diagram of the Kalman-Bucy filter.

The statistical assumptions about the initial state and the noises are

$$E[x(0)] = \hat{x}(0) \quad (9.2.7-1)$$

$$\text{cov}[x(0)] = P(0) \quad (9.2.7-2)$$

$$E[\tilde{v}(t)] = 0 \quad (9.2.7-3)$$

$$E[\tilde{v}(t)\tilde{v}(\tau)'] = \tilde{Q}(t)\delta(t - \tau) \quad (9.2.7-4)$$

$$E[\tilde{w}(t)] = 0 \quad (9.2.7-5)$$

$$E[\tilde{w}(t)\tilde{w}(\tau)'] = \tilde{R}(t)\delta(t - \tau) \quad (9.2.7-6)$$

and that these three sources of uncertainty are mutually uncorrelated.

## 9.2.8 Continuous-Time State Estimation — Summary

The LMMSE state estimator for a continuous-time linear dynamic system driven by white process noise, and whose state is observed in the presence of white measurement noise, has been derived.

This estimator, called the Kalman-Bucy filter, consists of a linear differential equation, similar to the static equation but driven by the innovation multiplied by the filter gain.

The filter gain is given in terms of the covariance matrix of the state estimate, which is the same as the covariance of the estimation error. This is obtained as the solution of a Riccati matrix differential equation.

This filter has been derived for a system that can be *time-varying* and with possibly *nonstationary* noises.

Conditions, under which for a time-invariant system with stationary noises the Riccati equation for the error covariance has a unique positive definite solution, have been given. Under these conditions the filter gain converges to a steady-state value, and the resulting steady-state filter is stable — the estimation error becomes stationary.

Similarly to the discrete time case, the continuous-time LMMSE estimator is the best among the class of linear estimators. If

- the initial state,
- the process noise stochastic process, and
- the measurement noise stochastic process

are all Gaussian, then this estimator yields the optimal MMSE estimate — the conditional mean of the state.

## 9.3 PREDICTION AND THE CONTINUOUS-DISCRETE FILTER

### 9.3.1 Prediction of the Mean and Covariance

Prediction is the estimation of the state at time  $t_2$  beyond the data interval, that is, based on data up to an earlier time  $t_1 < t_2$ ,

$$\hat{x}(t_2|t_1) = E[x(t_2)|Z^{t_1}] \quad (9.3.1-1)$$

where

$$Z^{t_1} \triangleq \{z(t), t \in [t_0, t_1]\} \quad (9.3.1-2)$$

and  $t_0$  denotes the initial time.

With the system

$$\dot{x}(t) = A(t)x(t) + D(t)\tilde{v}(t) \quad (9.3.1-3)$$

one has

$$x(t_2) = F(t_2, t_1)x(t_1) + \int_{t_1}^{t_2} F(t_2, \tau)D(\tau)\tilde{v}(\tau) d\tau \quad (9.3.1-4)$$

Applying the conditional expectation (9.3.1-1) on (9.3.1-4) yields the *predicted state*

$$\hat{x}(t_2|t_1) = F(t_2, t_1)\hat{x}(t_1) \quad (9.3.1-5)$$

in view of the fact that the process noise  $\tilde{v}(t)$  is zero mean and white.

The above is actually the solution of the differential equation (9.2.3-11) repeated below

$$\dot{\hat{x}}(t) = A(t)\hat{x}(t) \quad (9.3.1-6)$$

at  $t = t_2$  with initial condition  $\hat{x}(t_1)$ . If the transition matrix  $F(t_2, t_1)$  is not available in explicit form, then (9.3.1-5) cannot be used — instead one has to carry out numerical integration of (9.3.1-6) from  $t_1$  to  $t_2$  to obtain  $\hat{x}(t_2|t_1)$ .

The covariance associated with the prediction (9.3.1-5) — the ***prediction covariance*** — is

$$\begin{aligned} P(t_2|t_1) &\triangleq \text{cov}[x(t_2)|Z^{t_1}] \\ &= F(t_2, t_1)P(t_1)F(t_2, t_1)' + \int_{t_1}^{t_2} F(t_2, \tau)D(\tau)\tilde{Q}(\tau)D(\tau)'F(t_2, \tau)'d\tau \end{aligned} \quad (9.3.1-7)$$

where  $P(t_1)$  is the covariance of  $x(t_1)$  and  $\tilde{Q}$  is the intensity (power spectral density) of the process noise.

Similarly to the situation for the predicted state, which is given explicitly by (9.3.1-5) or is the solution of (9.3.1-6), the prediction covariance (9.3.1-7) is the solution of (9.2.3-12), repeated below

$$\dot{P}(t) = A(t)P(t) + P(t)A(t)' + D(t)\tilde{Q}(t)D(t)' \quad (9.3.1-8)$$

at  $t_2$  starting from  $t_1$  with initial condition  $P(t_1)$ .

### 9.3.2 The Various Types of Prediction

As in the discrete-time case, there are several types of prediction:

- (a) **Fixed-point prediction:** Prediction is done to the fixed point in time  $\tau$  (e.g., an intercept time) based on the data up to time  $t$ , where  $t$  is changing.
- (b) **Fixed-lead prediction:** Prediction is done for a fixed time  $T$  ahead (i.e., to  $t + T$ , where  $t$  varies and  $T$  is the lead).
- (c) **Fixed-interval prediction:** Prediction is done based on the data from the fixed interval up to  $t_1$  to times beyond it,  $t > t_1$ .

The equations for these three types of prediction are given below.

#### Fixed-Point Prediction

The predicted state is

$$\hat{x}(\tau|t) = F(\tau, t)\hat{x}(t) \quad (9.3.2-1)$$

It can be shown that the differential equation obeyed by this prediction is (see problem 9-3)

$$\begin{aligned} \frac{\partial}{\partial t}\hat{x}(\tau|t) &= -F(\tau, t)A(t)\hat{x}(t) + F(\tau, t)\dot{\hat{x}}(t) \\ &= -F(\tau, t)A(t)\hat{x}(t) + F(\tau, t)[A(t)\hat{x}(t) + L(t)\nu(t)] \\ &= F(\tau, t)L(t)\nu(t) \end{aligned} \quad (9.3.2-2)$$

(See also problem 9-4.)

### Fixed-Lead Prediction

This prediction for lead time  $T$  is given by the expression

$$\hat{x}(t+T|t) = F(t+T, t)\hat{x}(t) \quad (9.3.2-3)$$

The differential equation for this prediction is (see problem 9-5)

$$\frac{d}{dt}\hat{x}(t+T|t) = A(t+T)\hat{x}(t+T|t) + F(t+T, t)L(t)\nu(t) \quad (9.3.2-4)$$

### Fixed-Interval Prediction

The state prediction to  $t$  based on data up to  $t_1 < t$  is

$$\hat{x}(t|t_1) = F(t, t_1)\hat{x}(t_1) \quad (9.3.2-5)$$

and its propagation equation is (see problem 9-3)

$$\dot{\hat{x}}(t|t_1) = A(t)F(t, t_1)\hat{x}(t_1) = A(t)\hat{x}(t|t_1) \quad (9.3.2-6)$$

It can be easily shown that the covariance associated with (9.3.2-5)

$$P(t|t_1) \triangleq \text{cov}[x(t)|Z^{t_1}] \quad (9.3.2-7)$$

obeys the differential equation

$$\frac{d}{dt}P(t|t_1) = A(t)P(t|t_1) + P(t|t_1)A(t)' + D(t)\tilde{Q}(t)D(t)' \quad (9.3.2-8)$$

Note that this is the same as (9.2.3-12).

### 9.3.3 The Continuous-Discrete Filter

The state equation in *continuous time* is (as before, without input for simplicity)

$$\dot{x}(t) = A(t)x(t) + D(t)\bar{v}(t) \quad (9.3.3-1)$$

with  $\bar{v}(t)$  the process noise, zero mean and white

$$E[\bar{v}(t)] = 0 \quad (9.3.3-2)$$

$$E[\bar{v}(t)\bar{v}(\tau)'] = \tilde{Q}(t)\delta(t - \tau) \quad (9.3.3-3)$$

where  $\tilde{Q}(t)$  is the possibly time-varying intensity of the white noise.

The measurement equation in *discrete time* is

$$z(k) \triangleq z(t_k) = C(t_k)x(t_k) + w(t_k) \triangleq H(k)x(k) + w(k) \quad (9.3.3-4)$$

with the process noise  $w(t_k)$  a zero-mean white sequence

$$E[w(t_k)] = 0 \quad (9.3.3-5)$$

$$E[w(t_k)w(t_j)'] = R(k)\delta_{kj} \quad (9.3.3-6)$$

The process noise, the measurement noise and the initial state are mutually uncorrelated. The noises can be *nonstationary*.

## The Algorithm

The state estimation filter for this problem of continuous-time state equation and discrete time measurements, called the *continuous-discrete filter*, consists of the following:

1. Prediction stage: Between the times at which measurements are obtained, the state estimate and its covariance are propagated according to (9.2.3-11) and (9.2.3-12), yielding the corresponding predictions  $\hat{x}(k+1|k) \triangleq \hat{x}(t_{k+1}|t_k)$  and  $P(k+1|k) \triangleq P(t_{k+1}|t_k)$ .
2. Update stage: At the discrete times when the measurements are obtained, the state estimate and its covariance are updated according to the standard discrete-time Kalman filter update equations from Subsection 5.3.1.

## 9.4 DUALITY OF ESTIMATION AND CONTROL

### 9.4.1 The Two Problems

#### The Estimation Problem

Consider the LMMSE estimation problem for the following time-invariant system

$$\mathcal{S} : \begin{cases} \dot{\tilde{x}}(t) = Ax(t) + D\tilde{v}(t) \\ z(t) = Cx(t) + \tilde{w}(t) \end{cases} \quad (9.4.1-1)$$

with the following statistics

$$\text{cov}[x(0)] = P_0 \quad (9.4.1-2)$$

$$E[\tilde{v}(t)\tilde{v}(\tau)'] = Q\delta(t-\tau) \quad Q > 0 \quad (9.4.1-3)$$

$$E[\tilde{w}(t)\tilde{w}(\tau)'] = R\delta(t-\tau) \quad R > 0 \quad (9.4.1-4)$$

The initial state and the noises are mutually uncorrelated. The noise intensities  $Q$  and  $R$  are denoted in this section without a tilde.

#### The Control Problem

Consider the *linear-quadratic control problem (LQ)* for the *linear system*

$$\mathcal{S}^* : \begin{cases} \dot{y}(t) = -A'y(t) - C'u(t) \\ \psi(t) = D'y(t) \end{cases} \quad (9.4.1-5)$$

with the following *quadratic cost functional* (which maps the state and control time-functions into a scalar “cost”)

$$J(u, t_0, t_1) = y(t_1)'P_0y(t_1) + \int_{t_0}^{t_1} [\psi(t)'Q\psi(t) + u(t)'Ru(t)]dt \quad (9.4.1-6)$$

This cost is to be *minimized* by the function  $u(t)$ ,  $t \in [t_0, t_1]$ .

The **cost weighting matrices**  $Q$  and  $R$  in (9.4.1-6) are the same as the noise intensity matrices in (9.4.1-3) and (9.4.1-4), that is, positive definite. Relaxation of the requirement on  $Q$  to positive semidefinite is possible.

The system  $\mathcal{S}^*$  is called the **adjoint** of  $\mathcal{S}$ , since  $\mathcal{S}^*$  propagates backwards in time according to the system matrix of  $\mathcal{S}$  transposed.

## 9.4.2 The Solutions to the Estimation and the Control Problems

### The Estimation Solution

The optimal solution to the LMMSE estimation problem for system  $\mathcal{S}$  is

$$\begin{aligned}\dot{\hat{x}}(t) &= A\hat{x}(t) + L_e(t)[z(t) - C\hat{x}(t)] \\ &= [A - L_e(t)C]\hat{x}(t) + L_e(t)z(t)\end{aligned}\quad (9.4.2-1)$$

where the estimator (filter) gain is denoted as

$$L_e(t) = P(t)C'R^{-1} \quad (9.4.2-2)$$

The state estimation covariance follows from the Riccati equation

$$\dot{P}(t) = AP(t) + P(t)A' - P(t)C'R^{-1}CP(t) + DQD' \quad t > t_0 \quad (9.4.2-3)$$

with boundary condition at the *initial time*

$$P(t_0) = P_0 \quad (9.4.2-4)$$

### The Control Solution

The optimal solution to the LQ control problem for the adjoint system  $\mathcal{S}^*$  is the optimal feedback control

$$u^o(t) = -L_c(t)'y^o(t) \quad (9.4.2-5)$$

where the optimal state  $y^o(t)$  evolves according to the closed-loop equation

$$\dot{y}^o = -A'y^o(t) - C'u^o(t) = -[A - L_c(t)C]'y^o(t) \quad (9.4.2-6)$$

The control feedback gain in (9.4.2-5) is given by

$$L_c(t) = K(t)C'R^{-1} \quad (9.4.2-7)$$

and the **optimal cost matrix**  $K(t)$  follows from the Riccati equation

$$\dot{K}(t) = AK(t) + K(t)A' - K(t)C'R^{-1}CK(t) + DQD' \quad t < t_1 \quad (9.4.2-8)$$

with boundary condition at the *terminal time*  $t_1$

$$K(t_1) = P_0 \quad (9.4.2-9)$$

The optimized (minimized) value of the cost (9.4.1-6) is the **optimal cost**

$$J^o(t_0, t_1) \triangleq \min_u J(u) = y(t_0)'K(t_0)y(t_0) \quad (9.4.2-10)$$

The proof of the solution of the LQ problem can be found in, for example, [Bryson75].

## Duality of the Solutions

The optimal performance of the LMMSE state estimation problem starting from time  $t_0$  is given at time  $t > t_0$  by the (*optimal*) state covariance matrix  $P(t)$ .

The optimal performance of the LQ control problem starting at the arbitrary time  $t < t_1$  and running up to time  $t_1$  is given, similarly to (9.4.2-10), by

$$J^o(t, t_1) = y(t)' K(t) y(t) \quad (9.4.2-11)$$

that is, it is determined by the *optimal cost matrix*  $K(t)$ .

The following similarities between the solutions of the LMMSE estimation and the LQ control problems can be observed:

1. The covariance equation (9.4.2-3) of the estimation problem is identical to the cost matrix equation (9.4.2-8) of the control problem. The only difference is that, while the covariance equation propagates forward from the initial condition (9.4.2-4), the cost matrix equation propagates backwards from the terminal condition (9.4.2-9), which is, however, the same as (9.4.2-4).
2. The filter gain (9.4.2-2) has the same expression as the control feedback gain (9.4.2-7).
3. The closed-loop systems (9.4.2-1) from which the driving term is omitted and (9.4.2-6) are the adjoint of each other.

In view of the fact that their basic equations are the same, these two problems are *dual problems*<sup>2</sup>; and, as will be seen in the next subsection, their solutions exhibit similar properties under the same conditions.

### 9.4.3 Properties of the Solutions

The controllability matrix of system  $\mathcal{S}$  given in (9.4.1-1) is

$$\mathcal{Q}_C(\mathcal{S}) \triangleq [\tilde{D} \ A\tilde{D} \ \cdots \ A^{n_x-1}\tilde{D}] \quad (9.4.3-1)$$

where, similarly to (9.2.5-7),  $\tilde{D} = DQ^{1/2}$ , and this is the same as the observability matrix of system  $\mathcal{S}^*$  given in (9.4.1-5) for  $Q^{1/2}\psi$ , the vector whose norm appears in the control cost (9.4.1-6), i.e.,

$$\mathcal{Q}_C(\mathcal{S}) = Q_O(\mathcal{S}^*)' \quad (9.4.3-2)$$

The observability matrix of system  $\mathcal{S}$  is

$$\mathcal{Q}_O(\mathcal{S}) \triangleq [C' \ A'C' \ \cdots \ (A')^{n_x-1}C']' \quad (9.4.3-3)$$

---

<sup>2</sup>Thus, people who started their careers in control, but then turned to estimation, can be said to be “out of control”; however, in view of the duality of these two problems, they only appear to be out of control.

and this is the same as the controllability matrix of system  $\mathcal{S}^*$

$$\mathcal{Q}_O(\mathcal{S}) = \mathcal{Q}_C(\mathcal{S}^*)' \quad (9.4.3-4)$$

In view of the above equalities (9.4.3-2) and (9.4.3-4), the *dual problems* of LMMSE estimation and LQ control exhibit *similar properties* under the *same conditions*. These properties will be detailed next.

The solution to the estimation Riccati equation (9.4.2-3) with initial time  $t_0$  will be denoted as  $P_{t_0}(t)$ . Similarly, the solution to the control Riccati equation (9.4.2-8) with terminal time  $t_1$  will be denoted as  $K_{t_1}(t)$ .

With these notations, one has

- $P_{t_0}(\infty)$ : the steady-state covariance in the LMMSE estimation problem (i.e., with “infinite horizon”);
- $K_{t_1}(-\infty)$ : the LQ problem cost starting at  $t \rightarrow -\infty$  and going up to  $t_1$ , that is, the control problem over an infinite period of time (with “infinite horizon”).

The dual estimation and control problems have the following properties:

1. Under the equivalent conditions

$$\{\mathcal{S} \text{ is completely observable}\} \equiv \{\mathcal{S}^* \text{ is completely controllable}\}$$

one has

$$\lim_{t_0 \rightarrow -\infty} P_{t_0}(t) = \lim_{t \rightarrow \infty} P_{t_0}(t) \stackrel{\Delta}{=} P(\infty) = K(-\infty) \stackrel{\Delta}{=} \lim_{t \rightarrow -\infty} K_{t_1}(t) = \lim_{t_1 \rightarrow \infty} K_{t_1}(t) \quad (9.4.3-5)$$

where the common limit above is the not necessarily unique solution of the algebraic Riccati equation

$$AP + PA' + DQD' - PC'R^{-1}CP = 0 \quad (9.4.3-6)$$

2. If, in addition,

$$\{\mathcal{S} \text{ is completely controllable}\} \equiv \{\mathcal{S}^* \text{ is completely observable}\}$$

then

$$P_{t_0}(t) > 0 \quad \forall t > t_0 \quad (9.4.3-7)$$

$$K_{t_1}(t) > 0 \quad \forall t < t_1 \quad (9.4.3-8)$$

and the limit in (9.4.3-5) is the unique positive definite solution of (9.4.3-6).

3. Under the complete controllability and complete observability conditions, the eigenvalues of the *closed-loop system matrix*

$$A_c \stackrel{\Delta}{=} A - P(\infty)C'R^{-1}C \quad (9.4.3-9)$$

are in the left half-plane and thus the filter and the optimal feedback control system are asymptotically stable. This holds regardless of the eigenvalues of the open-loop system matrix  $A$ .

**Remark**

In view of (9.4.2-10), a *finite*  $K_{t_1}(-\infty)$  guarantees a *finite cost for the infinite horizon* problem; hence, the state and control vectors *converge to zero* since the integral of the sum of their norms is finite.

## 9.5 THE WIENER-HOPF PROBLEM

### 9.5.1 Formulation of the Problem

The **Wiener-Hopf problem** deals with the LMMSE estimation of a vector-valued stochastic process  $x(t)$  with mean and autocovariance

$$E[x(t)] = \bar{x}(t) \quad (9.5.1-1)$$

$$\text{cov}[x(t), x(\tau)] = V_{xx}(t, \tau) \quad (9.5.1-2)$$

based on the observed process  $z(t)$  with statistics

$$E[z(t)] = \bar{z}(t) \quad (9.5.1-3)$$

$$\text{cov}[z(t), z(\tau)] = V_{zz}(t, \tau) \quad (9.5.1-4)$$

The **cross-covariance** between the two processes is denoted as

$$\text{cov}[x(t), z(\tau)] = V_{xz}(t, \tau) \quad (9.5.1-5)$$

The estimator is of the form

$$\hat{x}(t) = \xi(t) + \int_0^t H(t, \tau)z(\tau)d\tau \quad (9.5.1-6)$$

and it should minimize the mean-square value of the norm of the error

$$E[\|\tilde{x}(t)\|^2] = \text{tr}\{\text{cov}[\tilde{x}(t)]\} \quad (9.5.1-7)$$

where the estimation error is defined as

$$\tilde{x}(t) \triangleq x(t) - \hat{x}(t) \quad (9.5.1-8)$$

Equation (9.5.1-6) has the form of an **affine transformation** of the function  $z(\tau)$  — a linear operator on this function plus another function.

### 9.5.2 The Wiener-Hopf Equation

The LMMSE estimator (9.5.1-6) is obtained by setting the following conditions on the estimation error (9.5.1-8):

1. The error should be zero mean (i.e., the estimator should be unbiased).

2. The error should be orthogonal to the observations.

The ***unbiasedness*** condition is

$$E[\hat{x}(t)] = \xi(t) + \int_0^t H(t, \tau) \bar{z}(\tau) d\tau = \bar{x}(t) \quad (9.5.2-1)$$

and it yields

$$\xi(t) = \bar{x}(t) - \int_0^t H(t, \tau) \bar{z}(\tau) d\tau \quad (9.5.2-2)$$

which leads to

$$\hat{x}(t) = \bar{x}(t) + \int_0^t H(t, \tau) [z(\tau) - \bar{z}(\tau)] d\tau \quad (9.5.2-3)$$

The ***orthogonality*** condition extended from random vectors to vector-valued random processes is

$$E[\tilde{x}(t) z(\theta)'] = 0 \quad \forall \theta \leq t \quad (9.5.2-4)$$

or

$$E[(x(t) - \hat{x}(t)) [z(\theta) - \bar{z}(\theta)']] = 0 \quad \forall \theta \leq t \quad (9.5.2-5)$$

Inserting (9.5.2-3) into the above, one has

$$E\left[\{x(t) - \bar{x}(t) - \int_0^t H(t, \tau) [z(\tau) - \bar{z}(\tau)] d\tau\} \{z(\theta) - \bar{z}(\theta)\}'\right] = 0 \quad (9.5.2-6)$$

which yields the ***Wiener-Hopf equation***

$$V_{xz}(t, \theta) - \int_0^t H(t, \tau) V_{zz}(\tau, \theta) d\tau = 0$$

(9.5.2-7)

## Remarks

This equation does not assume an explicit model that relates  $x$  to  $z$  — the modeling is implicit in their moments (means, autocovariances and the cross-covariance).

Equation (9.5.2-7) is an *integral equation* for the *impulse response*  $H(t, \tau)$  of the filter that is the optimal estimator for this problem. This equation cannot be solved for the general nonstationary case discussed above.

In the *stationary case*, the Wiener-Hopf equation becomes

$$V_{xz}(t - \theta) - \int_0^t H(t - \tau) V_{zz}(\tau - \theta) d\tau = 0 \quad \forall \theta \leq t \quad (9.5.2-8)$$

## Outline of the Solution

Using Fourier transforms, a modified version of this equation can be solved for *rational spectra* (power spectral densities of the processes involved), yielding the **Wiener filter**.

Assuming that the observation process starts at  $t = -\infty$  rather than at  $t = 0$ , (9.5.2-8) is replaced by

$$V_{xz}(t - \theta) = \int_{-\infty}^t H(t - \tau) V_{zz}(\tau - \theta) d\tau \quad \forall \theta \leq t \quad (9.5.2-9)$$

Denoting

$$\alpha \triangleq t - \tau \quad (9.5.2-10)$$

$$\beta \triangleq t - \theta \quad (9.5.2-11)$$

one has

$$\tau - \theta = -\alpha + \beta \quad (9.5.2-12)$$

and (9.5.2-9) becomes

$$V_{xz}(\beta) = \int_0^\infty H(\alpha) V_{zz}(\beta - \alpha) d\alpha \quad \forall \beta \geq 0 \quad (9.5.2-13)$$

If the lower limit of the integral in (9.5.2-13) is changed to  $-\infty$ , this equation can be solved via Fourier transforms for the impulse response  $H(\alpha)$ ; however, the resulting system will have poles in both the **left half-plane (LHP)** and **right half-plane (RHP)**; that is, it will be *noncausal*.<sup>3</sup> Using spectral factorization and partial fraction expansion, one can obtain a *causal system*, which is the desired filter, known as the Wiener filter.

## Brief Derivation of the Wiener Filter

The modified version of (9.5.2-13) is then

$$V_{xz}(\beta) = \int_{-\infty}^\infty H(\alpha) V_{zz}(\beta - \alpha) d\alpha \quad \forall \beta \geq 0 \quad (9.5.2-14)$$

Applying Fourier transform yields

$$H(s) = S_{xz}(s) S_{zz}(s)^{-1} \quad (9.5.2-15)$$

where  $s = j\omega$  is the Fourier transform argument,  $S_{xz}$  is the cross-spectrum between the signal of interest  $x$  and the observation  $z$ , and  $S_{zz}$  is the spectrum of  $z$ .

<sup>3</sup>The inverse Fourier transform of a transfer function with RHP poles yields an impulse response that is nonzero for negative times, i.e., *before* the impulse is applied (at time zero) — a noncausal system. This can also be seen as follows: allowing in (9.5.2-13)  $\alpha < 0$  implies allowing  $\tau > t$ ; that is, we “predict”  $x(t)$  using  $z(\tau)$  not only for  $\tau < t$  (which is all right), but also for  $\tau > t$ , which implies noncausality because in this case  $z(\tau)$  is not yet available at  $t$ .

Let the spectrum  $S_{zz}(s)$ , which has symmetrical pairs of zeros and poles (about the origin, because it is the Fourier transform of an autocorrelation, which is symmetrical about 0), be factorized as follows

$$S_{zz}(s) = \mathcal{S}_{zz}^+(s)\mathcal{S}_{zz}^-(s) = \mathcal{S}_{zz}^+(s)\mathcal{S}_{zz}^+(-s)' \quad (9.5.2-16)$$

where  $\mathcal{S}_{zz}^+(s)$  has all its zeros and poles in the LHP, and  $\mathcal{S}_{zz}^-(s)$  has all its zeros and poles in the RHP. The superscript  $+/-$  indicates that it is the Fourier transform of a time function over the positive/negative real line.

Then

$$H(s) = S_{xz}(s)[\mathcal{S}_{zz}^-(s)]^{-1}[\mathcal{S}_{zz}^+(s)]^{-1} \triangleq A(s)[\mathcal{S}_{zz}^+(s)]^{-1} \quad (9.5.2-17)$$

Note that only the last term above is guaranteed to correspond to a realizable (causal) system. The term  $A(s)$  will have a mix of LHP and RHP poles.

The final realizable solution is obtained by using a partial fraction expansion and grouping the resulting terms as follows

$$A(s) = A^+(s) + A^-(s) \quad (9.5.2-18)$$

where  $A^+(s)$  will contain all the LHP poles and  $A^-(s)$  will contain all the RHP poles of  $A(s)$ .

It can be shown (see, e.g., [Sage71]) that the optimal realizable solution — the **Wiener filter** — is

$$H_o(s) = \left\{ S_{xz}(s)[\mathcal{S}_{zz}^-(s)]^{-1} \right\}^+ [\mathcal{S}_{zz}^+(s)]^{-1} \quad (9.5.2-19)$$

where  $\{\cdot\}^+$  denotes the physically realizable part, i.e., with LHP poles.

### Example of a Wiener Filter

The two-dimensional system considered in (9.2.6-17) will be considered for the derivation of a Wiener filter that estimates its first state component. The frequency domain relationship between the driving noise  $\tilde{v}(t)$  and the state component of interest is

$$X_1(s) = \frac{1}{s^2 + 1} \tilde{V}(s) \quad (9.5.2-20)$$

and the observation is

$$z(t) = x_1(t) + \tilde{w}(t) \quad (9.5.2-21)$$

with power spectral densities (spectrum)

$$S_{\tilde{v}\tilde{v}} = \tilde{q} = 1 \quad S_{\tilde{w}\tilde{w}} = \tilde{r} = 3 \quad (9.5.2-22)$$

The spectrum of  $x$  is

$$S_{xx}(s) = \frac{S_{vv}(s)}{(s^2 + 1)^2} = \frac{1}{(s^2 + 1)^2} \quad (9.5.2-23)$$

The spectrum of  $z$  is

$$S_{zz}(s) = S_{xx}(s) + S_{ww}(s) = \frac{1}{(s^2 + 1)^2} + 3 = \frac{3s^4 + 6s^2 + 10}{(s^2 + 1)^2} \quad (9.5.2-24)$$

Factorizing the above yields the LHP part

$$S_{zz}^+(s) = \frac{\sqrt{3}(s + 0.2781 - j1.038)(s + 0.2781 + j1.038)}{s^2 + 1} \quad (9.5.2-25)$$

The term  $A(s)$  from (9.5.2-17) is

$$A(s) = \frac{1}{(s^2 + 1)^2} \frac{s^2 + 1}{\sqrt{3}(s - 0.2781 + j1.038)(s - 0.2781 - j1.038)} \quad (9.5.2-26)$$

which, after partial fraction expansion, yields the part with realizable poles as

$$A^+(s) = \frac{0.9648s + 0.465}{s^2 + 1} \quad (9.5.2-27)$$

Combining (9.5.2-25) and (9.5.2-27) into (9.5.2-19) yields

$$\begin{aligned} H_o(s) &= \frac{0.9648s + 0.465}{s^2 + 1} \frac{s^2 + 1}{\sqrt{3}(s + 0.2781 - j1.038)(s + 0.2781 + j1.038)} \\ &= \frac{0.557s + 0.155}{s^2 + 0.557s + 1.155} \end{aligned} \quad (9.5.2-28)$$

i.e., the resulting Wiener filter is the same as the steady-state Kalman-Bucy filter transfer function from (9.2.6-25).

## Wiener Filter vs. Kalman Filter

It can be shown (see, e.g., [Sage71]) that the Wiener filter is the *steady-state version* of the Kalman-Bucy filter. In view of this, and the fact that the Kalman-Bucy filter is for the more general *nonstationary* case, the Wiener filter will not be pursued further.

## 9.6 NOTES AND PROBLEMS

### 9.6.1 Bibliographical Notes

More details on continuous-time filtering, including Ito differential equations, can be found in [Sage71, Meditch69, Maybeck79, Maybeck82]. Explicit solutions of the filters for continuous-time WNA (double integrator) and WPA (triple integrator) motion models with position observation are presented in [Fitzgerald81]. The WNA model with position and velocity observations is

discussed in [Ekstrand85, Pachter87]. Numerical aspects of the Riccati equation are discussed in [Grewal93].

The linear-quadratic control problem is covered in many texts, for instance, [Bryson75, Meditch69, Maybeck79, Maybeck82].

The duality between estimation and control is treated in [Rhodes71].

More details on the Wiener filter can be found in [Sage71, Kailath81, Lewis86].

## 9.6.2 Problems

### 9-1 Derivative of the inverse of a matrix.

1. Prove (9.2.5-2). *Hint:* Use the fact that  $P(t)^{-1}P(t) = I$ .
2. Prove (9.2.5-3).

### 9-2 Riccati differential equation solution for the inverse covariance.

1. Show that if

$$\dot{x}(t) = A(t)x(t) \quad \text{and} \quad x(t)'x_a(t) = c$$

then  $x_a(t)$  is the solution of the *adjoint system*

$$\dot{x}_a(t) = -A(t)'x_a(t) \triangleq A_a(t)x_a(t)$$

and that the transition matrix of the adjoint system from  $t_0$  to  $t$  relates to the one of the original system as follows:

$$F_a(t, t_0) = F(t_0, t)' = [F(t, t_0)']^{-1}$$

2. Rewrite (9.2.5-3) in terms of the system matrix  $A_a(t)$  of the adjoint system and show that it has the solution

$$P(t)^{-1} = F_a(t, t_0)P(t_0)^{-1}F_a(t, t_0)' + \int_{t_0}^t F_a(t, \tau)C(\tau)'R(\tau)^{-1}C(\tau)F_a(t, \tau)'d\tau$$

3. Show that the above yields (9.2.5-4).

### 9-3 Fixed-point prediction evolution in continuous time.

Let  $F(t_2, t_1)$  be the transition matrix corresponding to the system matrix  $A(t)$ ; that is, the equation

$$\dot{x}(t) = A(t)x(t)$$

has the solution

$$x(t_2) = F(t_2, t_1)x(t_1)$$

1. Show that

$$\frac{\partial}{\partial t_2}F(t_2, t_1) = A(t_2)F(t_2, t_1)$$

$$\frac{\partial}{\partial t_1}F(t_2, t_1) = -F(t_2, t_1)A(t_1)$$

2. Prove (9.3.2-2).

**9-4 Fixed-point prediction prior moments.** Show that, for  $t_2 > t_1$ ,

$$E[\hat{x}(\tau|t_2)|Z^{t_1}] = \hat{x}(\tau|t_1)$$

$$\text{cov}[\hat{x}(\tau|t_2)|Z^{t_1}] = P(\tau|t_1) - P(\tau|t_2)$$

**9-5 Fixed-lead prediction evolution in continuous time.** Prove (9.3.2-4).

**9-6 State MSE matrix for continuous-time filter with arbitrary gain.** Show that, similarly to (5.2.3-18), the continuous-time Riccati equation (9.2.3-10) has the following alternative expression:

$$\dot{P}(t) = [A(t) - L(t)C(t)]P(t) + P(t)[A(t) - L(t)C(t)]' + L(t)\hat{R}(t)L(t)' + D(t)\hat{Q}(t)D(t)'$$

which holds for an *arbitrary filter gain*  $L(t)$ ; that is,  $P(t)$  is the state estimation MSE matrix.

*Hint:* Utilize the limiting form of the derivative of  $P(t) \triangleq E[\tilde{x}(t)\tilde{x}(t)']$  and use the differential forms (9.2.1-7) and (9.2.3-13).

**9-7 Amplitude, frequency and phase estimation of a sinusoid.** Consider the measurement of a sinusoid in noise

$$z(t) = a \sin(\omega t + \phi) + \tilde{v}(t) = h(x, t) + \tilde{v}(t) \quad 0 < t \leq T$$

where

$$x \triangleq [a \ \omega \ \phi]'$$

and  $\tilde{v}(t)$  is a zero-mean white Gaussian process with spectrum  $N_0$ .

1. Show that, if  $T \gg 2\pi/\omega$ , the maximum likelihood estimate of the parameter vector  $x$  is

$$\hat{x}_{ML} = \arg \min \left[ \int_0^T z(t)^2 dt + \frac{1}{2} a^2 T - \int_0^T z(t) a \sin(\omega t + \phi) dt \right]$$

*Hint:* Estimate  $x$  from

$$\zeta_i = \int_{t_{i-1}}^{t_i} z(t) dt$$

with

$$t_i - t_{i-1} = \Delta T \ll 2\pi/\omega$$

and  $0 = t_0 < t_1 < \dots < t_n = T$ , by passing to the limit  $\Delta T \rightarrow 0$ ,  $n \rightarrow \infty$ .

2. Show that the ML estimates are

$$\hat{\omega} = \arg \max_{\omega} \left| \int_0^T z(t) e^{j\omega t} dt \right|$$

$$\hat{\phi} = \frac{\int_0^T z(t) \cos \hat{\omega} t dt}{\int_0^T z(t) \sin \hat{\omega} t dt}$$

and

$$\hat{a} = \frac{2}{T} \int_0^T z(t) \sin(\hat{\omega} t + \hat{\phi}) dt$$

- 9-8 Kalman-Bucy filter with non-zero-mean noises.** Provide the KB filter equations for the case where  $E[\tilde{v}(t)] = \bar{v}(t)$  and  $E[\tilde{w}(t)] = \bar{w}(t)$  with the rest of the assumptions the same as in Subsection 9.2.7.
- 9-9 Wiener filter as the steady-state Kalman-Bucy filter.** Derive the Wiener filter for  $x_2$  in the system in (9.2.6-17). Show that it is the same as the corresponding steady state Kalman-Bucy filter.
- 9-10 The KF as a lowpass filter.** Plot the frequency responses of the steady-state filters from Subsection 9.2.6 and find their bandwidths.



# Chapter 10

## STATE ESTIMATION FOR NONLINEAR DYNAMIC SYSTEMS

### 10.1 INTRODUCTION

#### 10.1.1 Outline

This chapter deals with the estimation of the state of discrete-time nonlinear dynamic systems observed via nonlinear measurements.

The optimal estimator for this problem is presented in Section 10.2, and the difficulty in its implementation is discussed. Also a comparison is made between the optimal nonlinear estimator and the best linear estimator for a simple problem.

The suboptimal filter known as the *extended Kalman filter (EKF)* is derived in Section 10.3, and some of the problems encountered in its implementation are illustrated in an example. Methods of compensation for linearized filters are discussed in Sections 10.4 and 10.5.

Section 10.6 describes a numerical procedure that relies on the technique of dynamic programming to obtain the maximum a posteriori (MAP) estimate of the sequence of states of a nonlinear dynamic system — the *modal trajectory*.

Section 10.7 discusses the continuous-discrete estimation — when the system evolves in continuous time and the measurements are obtained in discrete time. The propagation of the state pdf between observations is done in this case according to the Fokker-Planck equation, and the updating is done with Bayes' formula.

#### 10.1.2 Nonlinear Estimation — Summary of Objectives

- Present the optimal discrete-time estimator.
- Discuss the difficulty in implementing it in practice due to

- memory requirements and
- computational requirements.
- Compare the optimal nonlinear estimator to the best linear estimator for a simple problem.
- Discuss numerical implementation of the optimal discrete-time estimator.
- Derive the suboptimal filter known as the extended Kalman filter.
- Illustrate some of the problems that can be encountered in its implementation.
- Present some methods of compensation for linearized filters.
- Describe an efficient numerical technique for MAP trajectory estimation in discrete time
- Discuss continuous-discrete filtering and the Fokker-Planck equation for the propagation of the pdf of the state of a nonlinear system.

## 10.2 ESTIMATION IN NONLINEAR STOCHASTIC SYSTEMS

### 10.2.1 The Model

The general state space model for discrete-time stochastic systems is of the form

$$x(k+1) = \mathbf{f}[k, x(k), u(k), v(k)] \quad (10.2.1-1)$$

where  $x$  is the state vector,  $u$  is the known input, and  $v$  is the process noise. The vector-valued function  $\mathbf{f}$  is, in general, time-varying and allows the noise to enter nonlinearly. Later, when dealing with nonlinear dynamic systems with additive noise,  $\mathbf{f}$  will be replaced by  $f$ .

The measurements are described by

$$z(k) = \mathbf{h}[k, x(k), w(k)] \quad (10.2.1-2)$$

where  $w$  is the measurement noise. The function  $\mathbf{h}$ , which is also vector-valued, can be time-varying. Later, when dealing with nonlinear measurements with additive noise,  $\mathbf{h}$  will be replaced by  $h$ .

*At least one* of the above two functions has to be nonlinear in order for the problem to be nonlinear.

The noise sequences will be assumed to be white with known pdf and mutually independent. The initial state is assumed to have a known pdf and also be independent of the noises.

More specific assumptions will be given later when the optimal estimator, which evaluates the pdf of the state conditioned on the observations, is derived.

### The Main Result

It will be shown that the *optimal nonlinear state estimator* consists of the computation of the *conditional pdf* of the state  $x(k)$  given all the information

available at time  $k$ : the prior information about the initial state, the intervening inputs and the measurements through time  $k$ .

### 10.2.2 The Optimal Estimator

The *information set* available at  $k$  is

$$I^k = \{Z^k, U^{k-1}\} \quad (10.2.2-1)$$

where  $Z^k$  is the cumulative set of observations through time  $k$ , which subsumes the initial information  $Z^0$  and  $U^{k-1}$  is the set of known inputs prior to time  $k$ . Since the information set (10.2.2-1) is growing with  $k$ , it is desirable to have a *nongrowing information state* that summarizes the past in a manner appropriate for the problem.

For a stochastic system, an *information state* is a function of the available information set that completely summarizes the past of the system in a probabilistic sense: It is a complete substitute for the past data in the pdf of any present and future quantity related to the system.

As shown in the next subsection, the conditional pdf

$$p_k \triangleq p[x(k)|I^k] \quad (10.2.2-2)$$

satisfies this requirement if the two noise sequences (process and measurement) are *white and mutually independent*.

The optimal estimator consists of the *recursive functional relationship* between the information states  $p_{k+1}$  and  $p_k$ , and is given by

$$p_{k+1} = \frac{1}{c} p[z(k+1)|x(k+1)] \int p[x(k+1)|x(k), u(k)] p_k dx(k) \quad (10.2.2-3)$$

where the integration is over the entire domain of  $x(k)$  and  $c$  is the normalization constant. The derivation of the above is carried out in the next subsection.

### Remarks

The following remarks are pertinent to the optimal estimation:

1. The implementation of (10.2.2-3) requires storage of a pdf, which is, in general, equivalent to an *infinite-dimensional vector*. Furthermore, one faces the problem of carrying out the integration numerically.
2. In spite of these difficulties, the use of the information state has the following advantages:
  - a. It can be approximated over a grid of points or by a piecewise analytical function.

- b. It yields directly the MMSE estimate of the current state — the conditional mean — and any other desired information pertaining to the current state (e.g., its conditional variance).
- 3. For linear systems with Gaussian noises and Gaussian initial state, the functional recursion (10.2.2-3) becomes the Kalman Filter. In this case one has a *finite-dimensional sufficient statistic* consisting of the state's conditional mean and covariance that summarize completely the past in a probabilistic sense.
- 4. If a system is linear but the noises and/or the initial condition are not Gaussian, then, in general, there is no simple sufficient statistic and the recursion (10.2.2-3) has to be used to obtain the optimal MMSE estimator.

A simple example that compares the optimum and the linear MMSE estimators for a linear problem with uniformly distributed random variables is presented in Subsection 10.2.4.

### 10.2.3 Proof of the Recursion of the Conditional Density of the State

The conditional density of  $x(k+1)$  can be written using Bayes' formula

$$\begin{aligned} p_{k+1} &\triangleq p[x(k+1)|I^{k+1}] = p[x(k+1)|I^k, z(k+1), u(k)] \\ &= \frac{1}{c} p[z(k+1)|x(k+1), I^k, u(k)] p[x(k+1)|I^k, u(k)] \end{aligned} \quad (10.2.3-1)$$

where  $c$  is the normalization constant.

If the measurement noise is white in the following sense:  $w(k+1)$  conditioned on  $x(k+1)$  has to be independent of  $w(j)$ ,  $j \leq k$  (i.e., state-dependent measurement noise is allowed) and of  $v(j)$ ,  $j \leq k$  (i.e., the two noise sequences have to be independent), then

$$p[z(k+1)|x(k+1), I^k, u(k)] = p[z(k+1)|x(k+1)] \quad (10.2.3-2)$$

Note that the control (known input) is irrelevant in the above conditioning. The above expression is the *likelihood function* of  $x(k+1)$  given  $z(k+1)$ .

For an arbitrary value of the control at  $k$ , one has the *state prediction pdf* (a function, rather than the *point prediction* — the predicted value — as in the linear case)

$$p[x(k+1)|I^k, u(k)] = \int p[x(k+1)|x(k), I^k, u(k)] p[x(k)|I^k, u(k)] dx(k) \quad (10.2.3-3)$$

The above equation, which is an immediate consequence of the total probability theorem, is known as the *Chapman-Kolmogorov equation*.

It will be assumed that the process noise sequence is white and independent of the measurement noises in the following sense:  $v(k)$  conditioned on  $x(k)$  has to be independent of  $v(j-1)$ ,  $w(j)$ ,  $j \leq k$  (i.e., state-dependent process noise

is allowed). The whiteness of  $v(k)$  is equivalent to requiring the state vector  $x(k)$  to be a Markov process.

Then  $I^k$  is irrelevant in the conditioning of the first term on the right-hand side of (10.2.3-3), that is,

$$p[x(k+1)|x(k), I^k, u(k)] = p[x(k+1)|x(k), u(k)] \quad (10.2.3-4)$$

The above is the **state transition pdf**. Furthermore, since the input  $u(k)$  enters the system after the realization of  $x(k)$  has occurred, one has

$$p[x(k)|I^k, u(k)] = p[x(k)|I^k] \triangleq p_k \quad (10.2.3-5)$$

Inserting (10.2.3-4) and (10.2.3-5) into (10.2.3-3), it follows that the state prediction pdf can be written as

$$p[x(k+1)|I^k, u(k)] = \int p[x(k+1)|x(k), u(k)] p_k dx(k) = \phi[k+1, p_k, u(k)] \quad (10.2.3-6)$$

where  $\phi$  is a transformation — an **operator** — that maps the function  $p_k$  into another function, namely, the state prediction pdf.

Using (10.2.3-2) and (10.2.3-6) in (10.2.3-1), the latter can be rewritten as another transformation  $\psi$  that maps  $p_k$  into  $p_{k+1}$

$$\boxed{p_{k+1} = \psi[k+1, p_k, z(k+1), u(k)]} \quad (10.2.3-7)$$

The above is the functional recursion for the information state rewritten explicitly in (10.2.2-3).

## The Information State

Next it will be shown that  $p_k$  is an **information state** according to the definition of the previous subsection, i.e., that the pdf of any future state at  $j > k$  depends *only* on  $p_k$  and the intervening controls. These controls (known inputs) are denoted as

$$U_k^{j-1} \triangleq \{u(i)\}_{i=k}^{j-1} \quad (10.2.3-8)$$

The proof will be done using the **total probability theorem** (1.4.10-10). For  $j > k$  one has

$$\begin{aligned} p[x(j)|I^k, U_k^{j-1}] &= \int p[x(j)|x(k), I^k, U_k^{j-1}] p[x(k)|I^k] dx(k) \\ &= \int p[x(j)|x(k), U_k^{j-1}] p_k dx(k) \\ &\triangleq \mu[j, p_k, U_k^{j-1}] \end{aligned} \quad (10.2.3-9)$$

where the whiteness of the process noise sequence and its independence from the measurement noises have been used again. In the above,  $\mu$  denotes the

transformation that maps  $p_k$  and the known inputs into the pdf of a future state  $x(j)$ .

Thus from (10.2.3-7) and (10.2.3-9) it follows that  $I^k$  is summarized by  $p_k$ .

Therefore, the *whiteness and mutual independence of the two noise sequences* is a sufficient condition for  $p_k$  to be an information state. It should be emphasized that the whiteness is the crucial assumption.

The above conditions are equivalent to the requirement that  $x(k)$  be an ***incompletely observed Markov process*** — a Markov process observed partially and/or in the presence of noise.

If, for example, the process noise sequence is not white, it is obvious that  $p_k$  does not summarize the past data. In this case the vector  $x$  is not a state anymore and it has to be augmented.

This discussion points out the reason why the formulation of stochastic estimation and control problems is done with *mutually independent white noise sequences*.

#### 10.2.4 Example of Linear vs. Nonlinear Estimation of a Parameter

Consider an unknown parameter  $x$  with a prior pdf uniform in the interval  $[x_1, x_2]$ , that is,

$$p(x) = \mathcal{U}[x; x_1, x_2] \triangleq \frac{1(x - x_1) - 1(x - x_2)}{x_2 - x_1} \quad (10.2.4-1)$$

where  $1(\cdot)$  denotes the unit step function.

A measurement is made

$$z = x + w \quad (10.2.4-2)$$

where  $w$  is independent of  $x$  and uniformly distributed within the interval  $[-a, a]$

$$p(w) = \mathcal{U}[w; -a, a] = \frac{1(w + a) - 1(w - a)}{2a} \quad (10.2.4-3)$$

#### The Optimal MMSE Estimator

First the optimal MMSE estimator (the conditional mean) is obtained. The conditional pdf of  $x$  given  $z$  is

$$p(x|z) = \frac{p(z|x)p(x)}{p(z)} \quad (10.2.4-4)$$

From (10.2.4-3) it follows that the likelihood function of  $x$  given  $z$  — the pdf of  $z$  conditioned on  $x$  — is

$$p(z|x) = \mathcal{U}[z; x - a, x + a] = \frac{1[z - (x - a)] - 1[z - (x + a)]}{2a} \quad (10.2.4-5)$$

Using (10.2.4-1) and (10.2.4-5) in (10.2.4-4) yields the *conditional pdf* of  $x$  given  $z$  as

$$\begin{aligned} p(x|z) &= \frac{\{1[x-(z-a)] - 1[x-(z+a)]\}[1(x-x_1) - 1(x-x_2)]}{2a(x_2-x_1)p(z)} \\ &= \mathcal{U}[x; (z-a) \vee x_1, (z+a) \wedge x_2] \end{aligned} \quad (10.2.4-6)$$

where the following notations are used

$$a \vee b \triangleq \max(a, b) \quad (10.2.4-7)$$

$$a \wedge b \triangleq \min(a, b) \quad (10.2.4-8)$$

The conditional pdf (10.2.4-6) is seen to be uniform in the interval, which is the intersection of the interval  $[x_1, x_2]$  from the prior and the interval  $[z-a, z+a]$  — the feasibility region of  $x$  given  $z$  — for obvious reasons.

The MMSE estimator of  $x$  given  $z$  (i.e., its *exact conditional mean*) is, from (10.2.4-6),

$$\hat{x}^{\text{MMSE}} = E[x|z] = \frac{(z-a) \vee x_1 + (z+a) \wedge x_2}{2} \quad (10.2.4-9)$$

Note that this estimator is a *nonlinear function* of the measurement  $z$ .

The *conditional variance* of this estimator is, for a given  $z$ , from (10.2.4-6) and (1.4.5-8), given by

$$P_{xx|z} = \text{var}(x|z) = \frac{[(z-a) \vee x_1 - (z+a) \wedge x_2]^2}{12} \quad (10.2.4-10)$$

Note that the accuracy of the optimal estimate, measured in terms of its variance, is *measurement-dependent* — it depends on the measurement  $z$ . This is a major difference between nonlinear estimators, like (10.2.4-9), and linear estimators.

## The LMMSE Estimator

The *linear* MMSE estimator of  $x$  in terms of  $z$  is

$$\hat{x}^{\text{LMMSE}} = \bar{x} + P_{xz}P_{zz}^{-1}(z - \bar{z}) \quad (10.2.4-11)$$

where

$$\bar{x} = E[x] = \frac{x_1 + x_2}{2} \quad (10.2.4-12)$$

$$\bar{z} = E[z] = \bar{x} \quad (10.2.4-13)$$

$$\begin{aligned} P_{xz} &= E[(x-\bar{x})(z-\bar{z})] = E[(x-\bar{x})(x-\bar{x}+w)] \\ &= E[(x-\bar{x})^2] = P_{xx} = \frac{(x_1 - x_2)^2}{12} \end{aligned} \quad (10.2.4-14)$$

$$\begin{aligned} P_{zz} &= E[(z - \bar{z})^2] = E[(x - \bar{x} + w)^2] \\ &= P_{xx} + P_{ww} = \frac{(x_1 - x_2)^2 + 4a^2}{12} \end{aligned} \quad (10.2.4-15)$$

The MSE corresponding to the estimator (10.2.4-11) is

$$\begin{aligned} P_{xx}^{\text{LMMSE}} &= P_{xx} - \frac{P_{xz}^2}{P_{zz}} = \frac{P_{xx}P_{zz} - P_{xz}^2}{P_{zz}} \\ &= \frac{P_{xx}(P_{xx} + P_{ww}) - P_{xx}^2}{P_{xx} + P_{ww}} \\ &= \frac{P_{xx}P_{ww}}{P_{xx} + P_{ww}} \end{aligned} \quad (10.2.4-16)$$

### Comparison of the Errors: Optimal vs. Best Linear

The comparison between the errors obtained from the *optimal* and *best linear* methods cannot be made between (10.2.4-16) and (10.2.4-10) since the latter is a function of the observations. It has to be made between (10.2.4-16), which is independent of  $z$ , and the *expected value* of (10.2.4-10), which is to be *averaged over all the possible measurements  $z$* .

The average covariance associated with the *optimal estimate* is

$$P_{xx}^{\text{OPT}} = E[P_{xx|z}] = \int P_{xx|z} p(z) dz \quad (10.2.4-17)$$

where, in this example,  $p(z)$  is the convolution of the densities (10.2.4-1) and (10.2.4-3).

Table 10.2.4-1 illustrates the comparison of these two estimators' variances for some numerical values. The interval over which  $x$  is uniformly distributed is  $x_2 - x_1 = 1$ , that is, its "prior" variance was  $1/12 = 0.0833$ . The measurement ranged from very accurate for  $a \ll 1$ , to very inaccurate — practically noninformative — for  $a \gg 1$ .

### Conclusion

As can be seen from Table 10.2.4-1, the benefit from the nonlinear estimation over the linear one is disappointingly modest in this case: It ranges from negligible — 1% for the inaccurate measurement considered — to 6% for the accurate measurement; its maximum, which occurs in the midrange, is about 15% in variance. This latter improvement in variance translates into a 7% improvement in standard deviation (see problem 10-2).

No general conclusions can be drawn from the above toy/academic example — each problem requires its own evaluation. Nevertheless, it is an indication that in some problems the benefit from nonlinear estimation can be quite limited.

**Table 10.2.4-1:** Comparison of best linear and optimal estimators.

$a$	$P_{xx}^{\text{LMMSE}}$	$P_{xx}^{\text{OPT}}$
0.05	8.25E-04	7.64E-04
0.1	3.2E-03	2.81E-03
0.2	11.49E-03	9.5371E-03
0.5	4.16E-02	3.5871E-02
0.8	5.99E-02	5.4693E-02
1	6.66E-02	6.2510E-02
2	7.84E-02	7.6387E-02
5	8.25E-02	8.1249E-02

### 10.2.5 Estimation in Nonlinear Systems with Additive Noise

Consider the system with dynamics

$$x(k+1) = f[k, x(k)] + v(k) \quad (10.2.5-1)$$

where, for simplicity, it is assumed that there is no input/control, and the noise is assumed additive, with pdf  $p_{v(k)}[v(k)]$ , and white. The pdf is now identified by the appropriate subscript, since its argument will not always be the random variable it pertains to.

The measurement is

$$z(k) = h[k, x(k)] + w(k) \quad (10.2.5-2)$$

where the measurement noise is additive, with pdf  $p_{w(k)}[w(k)]$ , white, and independent from the process noise.

The initial state has the prior pdf  $p[x(0)]$  and is assumed to be independent from the two noise sequences.

### Prediction

Assuming that

$$p_k \triangleq p[x(k)|Z^k] \quad (10.2.5-3)$$

is available on a grid with  $n_k$  points  $\{x(k)_j\}_{j=1}^{n_k}$ , the predicted pdf follows from (10.2.3-6) as

$$p[x(k+1)|Z^k] = \int p[x(k+1)|x(k)]p[x(k)|Z^k] dx(k) \quad (10.2.5-4)$$

In view of the additivity of the process noise in (10.2.5-1),

$$p[x(k+1)|x(k)] = p_{v(k)}\{x(k+1) - f[k, x(k)]\} \quad (10.2.5-5)$$

and thus (10.2.5-4) becomes the following nonlinear convolution

$$p[x(k+1)|Z^k] = \int p_{v(k)}\{x(k+1) - f[k, x(k)]\}p[x(k)|Z^k] dx(k) \quad (10.2.5-6)$$

If the grid points for the state at  $k+1$  are  $\{x(k+1)_i\}_{i=1}^{n_{k+1}}$ , the numerical implementation of the above integral is

$$p[x(k+1)_i|Z^k] = \sum_{j=1}^{n_k} \mu_j p_{v(k)}\{x(k+1)_i - f[k, x(k)_j]\}p[x(k)_j|Z^k] \quad (10.2.5-7)$$

where  $\mu_j$  are the weighting coefficients of the numerical integration scheme used.

## Update

The update is, using (10.2.3-1) and (10.2.3-2), given by

$$p[x(k+1)|Z^{k+1}] = \frac{1}{c} p[z(k+1)|x(k+1)]p[x(k+1)|Z^k] \quad (10.2.5-8)$$

where  $c$  is the normalization constant. The numerical implementation of the above is simple,

$$p[x(k+1)_i|Z^{k+1}] = \frac{1}{c} p[z(k+1)|x(k+1)_i]p[x(k+1)_i|Z^k] \quad (10.2.5-9)$$

but it requires the evaluation of the normalization constant

$$c = \int p[z(k+1)|x(k+1)]p[x(k+1)|Z^k] dx(k+1) \quad (10.2.5-10)$$

which is done numerically as

$$c = \sum_{i=1}^{n_{k+1}} \mu_i p[z(k+1)|x(k+1)_i]p[x(k+1)_i|Z^k] \quad (10.2.5-11)$$

## Remarks

The number of grid points for a state of dimension  $n_x$ , with  $m$  points per state component, is  $m^{n_x}$ . This **curse of dimensionality** is the major stumbling block in making numerical techniques practical due to the ensuing heavy storage and computational requirements.

Another issue is the selection of the grid, which evolves in time. The points have to cover a region in the state space outside which the probability mass is negligible.

For details and extensive references on numerical techniques, including series expansion approaches, see [Sorenson88].

### 10.2.6 Optimal Nonlinear Estimation — Summary

Given a system described by

1. dynamic equation perturbed by white process noise and
2. measurement equation perturbed by white noise independent of the process noise,

with at least one of these equations nonlinear, the estimation of the system's state consists then of the calculation of its pdf conditioned on the entire available information: the observations, the initial state information and the past inputs.

This conditional pdf has the property of being an *information state*, that is, it summarizes probabilistically the past of the system.

The optimal nonlinear estimator for such a discrete-time stochastic dynamic system consists of a recursive functional relationship for the state's conditional pdf. This conditional pdf then yields the MMSE estimator for the state — the conditional mean of the state.

This nonlinear estimator has to be used to obtain the conditional mean of the state unless the system is *linear Gaussian*: Both the dynamic and the measurement equations are linear and all the noises and the initial state are Gaussian.

If the system is linear Gaussian, this functional recursion becomes the recursion for the conditional mean and covariance.

Unlike the linear case, in the nonlinear case, the accuracy of the estimator is *measurement-dependent*.

The numerical implementation of the nonlinear estimator on a set of grid points in the state space can be very demanding computationally — it suffers from the curse of dimensionality: The memory and computational requirements are exponential in the dimension of the state.<sup>1</sup>

## 10.3 THE EXTENDED KALMAN FILTER

### 10.3.1 Approximation of the Nonlinear Estimation Problem

In view of the very limited feasibility of the implementation of the optimal filter, which consists of a functional recursion, suboptimal<sup>2</sup> algorithms are of interest.

The recursive calculation of the sufficient statistic consisting of the conditional mean and variance in the linear-Gaussian case is the simplest possible state estimation filter. As indicated earlier, in the case of a linear system with non-Gaussian random variables the same simple recursion yields an approximate mean and variance.

---

<sup>1</sup>We feel that (in most cases) complexity is more a vice than a virtue.

<sup>2</sup>This is a common euphemism for nonoptimal.

A framework similar to the one from linear systems is desirable for a nonlinear system.<sup>3</sup> Such an estimator, called the ***extended Kalman filter (EKF)***, can be obtained by a series expansion of the nonlinear dynamics and of the measurement equations.

The (first-order) EKF is based on

- Linearization — first order series expansion — of the nonlinearities (in the dynamic and/or the measurement equation)
- LMMSE estimation

The second-order EKF relies on a second-order expansion, that is, it includes second-order correction terms.

### Modeling Assumptions

Consider the system with dynamics

$$x(k+1) = f[k, x(k)] + v(k) \quad (10.3.1-1)$$

where, for simplicity, it is assumed that there is no control, and the noise is assumed additive, zero mean, and white

$$E[v(k)] = 0 \quad (10.3.1-2)$$

$$E[v(k)v(j)'] = Q(k)\delta_{kj} \quad (10.3.1-3)$$

The measurement is

$$z(k) = h[k, x(k)] + w(k) \quad (10.3.1-4)$$

where the measurement noise is additive, zero mean, and white

$$E[w(k)] = 0 \quad (10.3.1-5)$$

$$E[w(k)w(j)'] = R(k)\delta_{kj} \quad (10.3.1-6)$$

and uncorrelated with the process noise.

The initial state, with estimate  $\hat{x}(0|0)$  — an approximate conditional mean — and the associated covariance matrix  $P(0|0)$ , is assumed to be uncorrelated with the two noise sequences.

Similarly to the linear case, it will be assumed that one has the estimate at time  $k$ , an *approximate conditional mean*

$$\hat{x}(k|k) \approx E[x(k)|Z^k] \quad (10.3.1-7)$$

and the *associated covariance matrix*  $P(k|k)$ .

Strictly speaking,  $P(k|k)$  is the MSE matrix rather than the covariance matrix in view of the fact that  $\hat{x}(k|k)$  is not the exact conditional mean.

Note that (10.3.1-7) implies that the estimation error is *approximately zero mean*. Another assumption that will be made is that the third-order moments of the estimation error are approximately zero — this is exact in the case of zero-mean Gaussian random variables.

---

<sup>3</sup>The most common use of linear filtering is for nonlinear systems [Sorenson85].

### 10.3.2 Derivation of the EKF

#### State Prediction

To obtain the predicted state  $\hat{x}(k+1|k)$ , the nonlinear function in (10.3.1-1) is expanded in Taylor series around the latest estimate  $\hat{x}(k|k)$  with terms up to first or second order to yield the first- or second-order EKF, respectively. The second order filter is presented from which the first-order one follows as a particular case.

The vector *Taylor series expansion* of (10.3.1-1) up to second-order terms is

$$\begin{aligned} x(k+1) &= f[k, \hat{x}(k|k)] + f_x(k)[x(k) - \hat{x}(k|k)] \\ &\quad + \frac{1}{2} \sum_{i=1}^{n_x} e_i [x(k) - \hat{x}(k|k)]' f_{xx}^i(k) [x(k) - \hat{x}(k|k)] \\ &\quad + HOT + v(k) \end{aligned} \quad (10.3.2-1)$$

where  $e_i$  is the  $i$ th  $n_x$ -dimensional **Cartesian basis vector** ( $i$ th component unity, the rest zero),

$$f_x(k) \triangleq [\nabla_x f(k, x)']'|_{x=\hat{x}(k|k)} \triangleq \frac{\partial f}{\partial x} \quad (10.3.2-2)$$

is the Jacobian of the vector  $f$ , evaluated at the latest estimate of the state. The last notation above, while used in the literature, is symbolic, since it is based on the scalar notation, and it will be avoided. Similarly,

$$f_{xx}^i(k) \triangleq [\nabla_x \nabla_x' f^i(k, x)]|_{x=\hat{x}(k|k)} \triangleq \frac{\partial^2 f^i}{\partial x^2} \quad (10.3.2-3)$$

is the Hessian of the  $i$ th component of  $f$  and HOT represents the *higher-order terms*, which we shall not touch — these terms will be neglected.

Note that, given the data  $Z^k$ , both the above Jacobian and the Hessian are *deterministic quantities* — only  $x(k)$  and  $v(k)$  are random variables in (10.3.2-1) in this case. These properties are needed when taking conditional expectations next.

The *predicted state* to time  $k+1$  from time  $k$  is obtained by taking the expectation of (10.3.2-1) conditioned on  $Z^k$  and neglecting the HOT

$$\hat{x}(k+1|k) = f[k, \hat{x}(k|k)] + \frac{1}{2} \sum_{i=1}^{n_x} e_i \text{tr}[f_{xx}^i(k) P(k|k)] \quad (10.3.2-4)$$

where use has been made of (1.4.15-1) for the second-order term. The first-order term in (10.3.2-1) is, in view of (10.3.1-7), (approximately) zero mean and thus vanishes.

The state prediction error is obtained by subtracting (10.3.2-4) from (10.3.2-1) to yield

$$\begin{aligned}\tilde{x}(k+1|k) &= f_x(k)\tilde{x}(k|k) + \frac{1}{2} \sum_{i=1}^{n_x} e_i \{ \tilde{x}(k|k)' f_{xx}^i(k) \tilde{x}(k|k) \\ &\quad - \text{tr}[f_{xx}^i(k) P(k|k)] \} + v(k)\end{aligned}\quad (10.3.2-5)$$

where the HOT have already been dropped.

Multiplying the above with its transpose and taking the expectation conditioned on  $Z^k$  yields the **state prediction covariance** (actually, the **MSE matrix**)

$$\boxed{\begin{aligned}P(k+1|k) &= f_x(k)P(k|k)f_x(k)' \\ &\quad + \frac{1}{2} \sum_{i=1}^{n_x} \sum_{j=1}^{n_x} e_i e_j' \text{tr}[f_{xx}^i(k) P(k|k) f_{xx}^j(k) P(k|k)] + Q(k)\end{aligned}}\quad (10.3.2-6)$$

where use has been made of identity (1.4.15-8) and of the approximation that the third-order moments are zero.

The state prediction (10.3.2-4) includes the second-order “correction” term. This is dropped in the first order EKF.

The prediction covariance (10.3.2-6) contains a fourth-order term obtained from the squaring of the second order term in (10.3.2-5).

The first-order version of (10.3.2-6) is the same as (5.2.3-5) in the linear filter — the Jacobian  $f_x(k)$  plays the role of the transition matrix  $F(k)$ .

## Measurement Prediction

Similarly, the **predicted measurement** is, for the second-order filter

$$\boxed{\hat{z}(k+1|k) = h[k+1, \hat{x}(k+1|k)] + \frac{1}{2} \sum_{i=1}^{n_z} e_i \text{tr}[h_{xx}^i(k+1) P(k+1|k)]}\quad (10.3.2-7)$$

where  $e_i$  denotes here the  $i$ th  $n_z$ -dimensional Cartesian basis vector.

The **measurement prediction covariance** or **innovation covariance** or **residual covariance** — really MSE matrix — is

$$\boxed{\begin{aligned}S(k+1) &= h_x(k+1)P(k+1|k)h_x(k+1)' \\ &\quad + \frac{1}{2} \sum_{i=1}^{n_z} \sum_{j=1}^{n_z} e_i e_j' \text{tr}[h_{xx}^i(k+1) P(k+1|k) h_{xx}^j(k+1) P(k+1|k)] + R(k+1)\end{aligned}}\quad (10.3.2-8)$$

where the Jacobian of  $h$  is

$$h_x(k+1) \triangleq [\nabla_x h(k+1, x)']' \Big|_{x=\hat{x}(k+1|k)} \triangleq \frac{\partial h}{\partial x} \quad (10.3.2-9)$$

and the Hessian of its  $i$ th component is

$$h_{xx}^i(k+1) \triangleq [\nabla_x \nabla_x' h^i(k+1, x)] \Big|_{x=\hat{x}(k+1|k)} \triangleq \frac{\partial^2 h^i}{\partial x^2} \quad (10.3.2-10)$$

## Modifications for the First-Order EKF

The second order “correction” term in (10.3.2-7) and the corresponding fourth-order term in (10.3.2-8) are dropped in the first-order EKF. The first-order version of (10.3.2-8) is the same as in the linear filter — the Jacobian  $h_x(k)$  plays the role of the measurement matrix  $H(k)$ .

### State Update

The expression of the filter gain, the update equation for the state and its covariance are identical to those from the linear filter, given by (5.2.3-11) to (5.2.3-15).

### 10.3.3 Overview of the EKF Algorithm

Figure 10.3.3-1 presents the flowchart of the extended Kalman filter (first-order version).

The main difference from the Kalman Filter, presented in Fig. 5.2.4-1, is the evaluation of the Jacobians of the state transition and the measurement equations. Due to this, the covariance computations are not decoupled anymore from the state estimate calculations and cannot be done, in general, offline.

The linearization (evaluation of the Jacobians) can be done, as indicated here, at the latest state estimate for  $F$  and the predicted state for  $H$ . Alternatively, it can be done along a *nominal trajectory* — a deterministic precomputed trajectory based on a certain scenario — which allows offline computation of the gain and covariance sequence.

### Stability of the EKF

The sufficient conditions for the stability of the KF are not necessarily sufficient for the EKF. The reason is that its inherent approximations can lead to *divergence of the filter* — unbounded estimation errors.

### A Cautionary Note about the EKF

It should be pointed out that the use of the series expansion in the state prediction and/or in the measurement prediction has the potential of introducing unmodeled errors that violate some basic assumptions about the prediction errors. These assumptions, which are implicit in the EKF, concern their moments, namely:

1. The prediction errors are zero mean (unbiased).
2. The prediction errors have covariances equal to the ones computed by the algorithm.

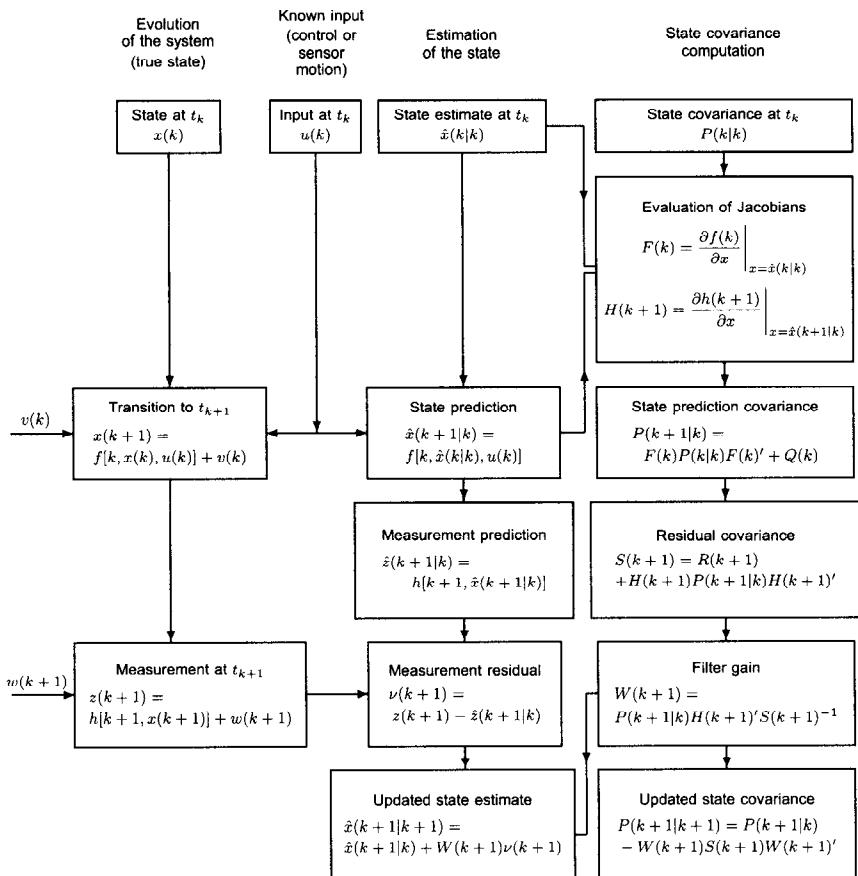


Figure 10.3.3-1: Flowchart of the EKF (one cycle).

In general, a nonlinear transformation will introduce a bias and the covariance calculation based on a series expansion is not always accurate. This might be the case with the predictions (10.3.2-4) and (10.3.2-7) and the covariances (10.3.2-6) and (10.3.2-8).

There is no guarantee that even the second-order terms can compensate for such errors. Also, the fact that these expansions rely on Jacobians (and Hessians in the second-order case) that are evaluated at the estimated or predicted state rather than the exact state (which is unavailable) can cause errors.

Thus it is even more important for an EKF to undergo *consistency testing*. The tests are the same as those discussed for the KF. Only if the bias is negligible with respect to the filter-calculated error variances and the mean-square errors are consistent with these variances is the filter consistent, and its results can be trusted.

In practice, if the initial error and the noises are not too large, then the EKF performs well. The actual limits of successful use of the linearization techniques implicit in the EKF can be obtained only via extensive Monte Carlo simulations for consistency verification.

#### 10.3.4 An Example: Tracking with an Angle-Only Sensor

A platform with a sensor moves in a plane according to the discrete-time equations

$$x_p(k) = \bar{x}_p(k) + \Delta x_p(k) \quad k = 0, 1, \dots, 20 \quad (10.3.4-1)$$

$$y_p(k) = \bar{y}_p(k) + \Delta y_p(k) \quad k = 0, 1, \dots, 20 \quad (10.3.4-2)$$

where  $\bar{x}_p(k)$  and  $\bar{y}_p(k)$  are the average platform position coordinates and the perturbations (vibrations or jitter)  $\Delta x_p(k)$  and  $\Delta y_p(k)$  are assumed to be mutually independent zero-mean Gaussian white noise sequences with variances  $r_x = 1$  and  $r_y = 1$ , respectively.

The average (unperturbed) platform motion is assumed to be horizontal with constant velocity. Its position as a function of the discrete time  $k$  is

$$\bar{x}_p(k) = 4k \quad (10.3.4-3)$$

$$\bar{y}_p(k) = 20 \quad (10.3.4-4)$$

A target moves on the  $x$ -axis according to

$$x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1/2 \\ 1 \end{bmatrix} v(k) \quad (10.3.4-5)$$

where

$$x(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \quad (10.3.4-6)$$

with  $x_1$  denoting the position and  $x_2$  denoting the velocity of the target.

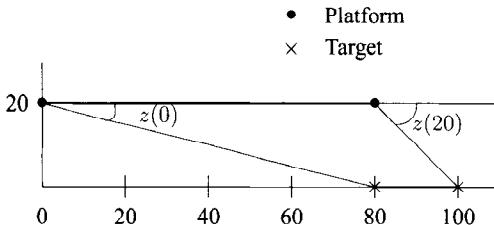


Figure 10.3.4-1: Platform and target motion.

The initial condition is

$$x(0) = \begin{bmatrix} 80 \\ 1 \end{bmatrix} \quad (10.3.4-7)$$

and the process noise  $v(k)$  is zero mean white with variance  $q = 10^{-2}$ .

The sensor measurement is

$$z(k) = h[x_p(k), y_p(k), x_1(k)] + w_s(k) \quad (10.3.4-8)$$

where

$$h[\cdot] = \tan^{-1} \frac{y_p(k)}{x_1(k) - x_p(k)} \quad (10.3.4-9)$$

is the angle between the horizontal and the line of sight from the sensor to the target, and the sensor noise  $w_s(k)$  is zero mean white Gaussian with variance  $r_s = (3^\circ)^2$ . The sensor noise is assumed independent of the sensor platform perturbations.

The estimation of the target's state is to be done using the measurements (10.3.4-8) and the knowledge of the unperturbed platform location (10.3.4-3) and (10.3.4-4).

Figure 10.3.4-1 illustrates the average motion of the sensor platform and the target.

The platform location perturbations induce additional errors in the measurements. The effect of these errors is evaluated by expanding the nonlinear measurement function  $h$  about the average platform position

$$\begin{aligned} z(k) &\approx h[\bar{x}_p(k), \bar{y}_p(k), x_1(k)] + \frac{\partial h}{\partial x_p} \Delta x_p(k) + \frac{\partial h}{\partial y_p} \Delta y_p(k) + w_s(k) \\ &\triangleq h[\bar{x}_p(k), \bar{y}_p(k), x_1(k)] + w(k) \end{aligned} \quad (10.3.4-10)$$

where the partials of  $h$  are evaluated at  $\bar{x}_p(k)$ ,  $\bar{y}_p(k)$ , and  $x_1(k)$ , while the HOT have been neglected.

The last three terms in (10.3.4-10) form the *equivalent measurement noise*  $w(k)$ , with variance

$$E[w(k)^2] \triangleq r(k) = \frac{\bar{y}_p(k)^2 r_x + [x_1(k) - \bar{x}_p(k)]^2 r_y}{\{|x_1(k) - \bar{x}_p(k)|^2 + \bar{y}_p(k)^2\}^2} + r_s \quad (10.3.4-11)$$

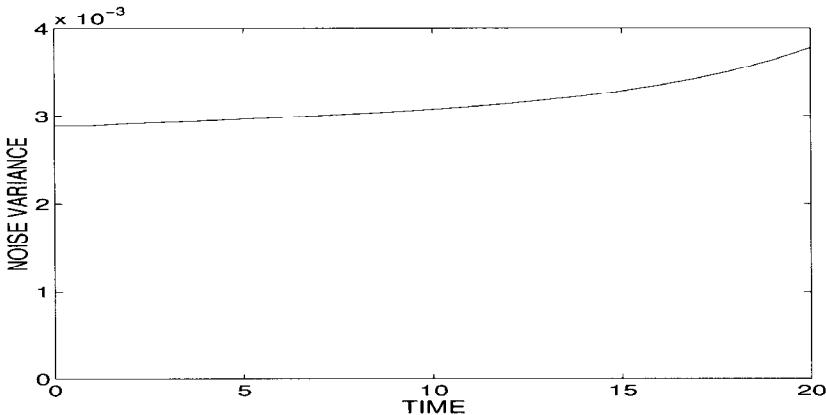


Figure 10.3.4-2: Variance of the equivalent measurement noise.

The above follows from the explicit evaluation of the partials in (10.3.4-10) and noting that the random variables  $\Delta x_p$ ,  $\Delta y_p$ , and  $w_s$  are mutually independent. Note that the variance (10.3.4-11) is time-varying. This variance (in rad<sup>2</sup>) is plotted in Fig. 10.3.4-2.

The state  $x$  is to be estimated based on the measurement equation (10.3.4-10) with the knowledge of only the average platform motion.

The initialization of the filter is done as follows. Based on Fig. 10.3.4-1 one can write the *inverse transformation* from the angle  $z$  to the position  $x_1$  as

$$\hat{x}_1(k) = g[z(k), \bar{x}_p(k), \bar{y}_p(k)] \triangleq \bar{x}_p(k) + \frac{\bar{y}_p(k)}{\tan z(k)} \quad k = 0, 1 \quad (10.3.4-12)$$

By differencing, one can obtain

$$\hat{x}_2(1) = \frac{\hat{x}_1(1) - \hat{x}_1(0)}{T} \quad (10.3.4-13)$$

where  $T = 1$  is the sampling period.

The variance associated with the above velocity estimate is

$$P_{22}(1|1) = \frac{P_{11}(1) + P_{11}(0)}{T^2} = P_{11}(1) + P_{11}(0) \quad (10.3.4-14)$$

where  $P_{11}(k)$ ,  $k = 0, 1$ , are the variances associated with (10.3.4-12).

Note, however, that the large angle measurement variance ( $r_s \approx 50$  mrad)<sup>2</sup>, when multiplied by the range ( $\approx 80$ ) squared, yields  $P_{11}(0) \approx P_{11}(1) \approx 4^2$ . In this situation, assuming (from a priori knowledge of the target) that  $|x_2| \leq 2$ , one is better off with the initial velocity estimate

$$\hat{x}_2(0|0) = 0 \quad (10.3.4-15)$$

and associated variance

$$P_{22}(0|0) = 1 \quad (10.3.4-16)$$

The off-diagonal term  $P_{12}(0|0)$  is taken as zero.

The initial position estimate is then  $\hat{x}_1(0|0)$ , as given by (10.3.4-12) for  $k = 0$ , and the variance associated with it is denoted as  $P_{11}(0|0)$ . This variance is obtained (approximately) by a first-order expansion of (10.3.4-12).

The error in the initial position estimate (10.3.4-12) is

$$\tilde{x}_1(0|0) = \frac{\partial g}{\partial \bar{x}_p} \Delta x_p + \frac{\partial g}{\partial \bar{y}_p} \Delta y_p + \frac{\partial g}{\partial z} \Delta z \quad (10.3.4-17)$$

where  $\Delta z$  is the error in the angle measurement, i.e.,  $w_s(0)$ , the sensor measurement noise.

Evaluating the partials in (10.3.4-17) yields

$$\tilde{x}_1(0|0) = \Delta x_p + \frac{1}{\tan z} \Delta y_p - \frac{\bar{y}_p}{\sin^2 z} \Delta z \quad (10.3.4-18)$$

from which

$$P_{11}(0|0) = r_x + \frac{1}{\tan^2 z} r_y + \frac{y_p^2}{\sin^4 z} r_s \quad (10.3.4-19)$$

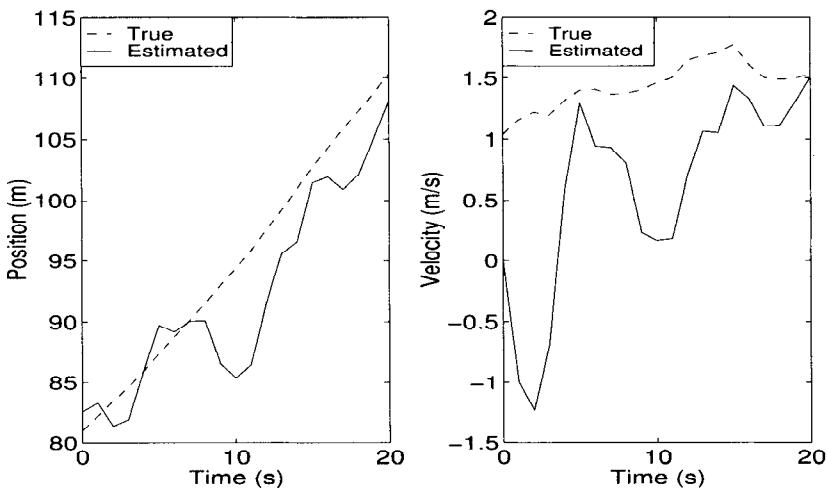
## Two Options for Implementation

In the problem considered here, as in many practical problems, the target motion is naturally modeled as linear in Cartesian coordinates, while the measurements are in polar coordinates (angle only in the present case). The nonlinear function  $h$  relates the state to the measurements. The following two options are available for implementation of the state estimation filter:

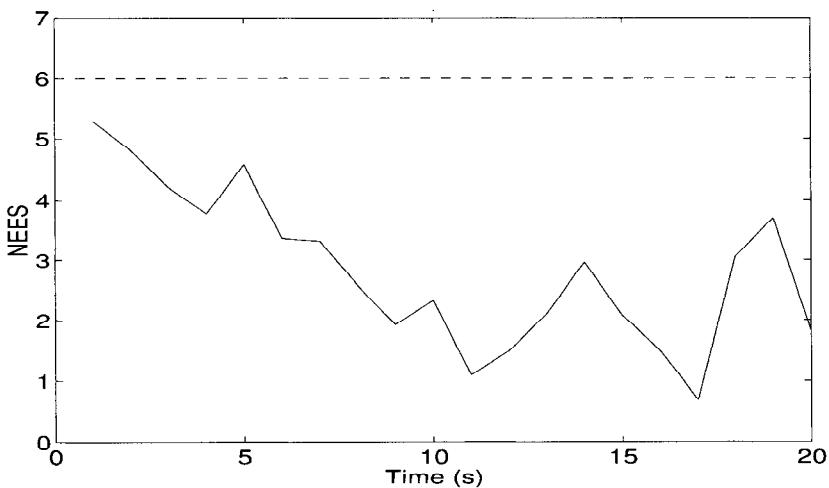
1. By using the inverse transformation  $\zeta \triangleq h^{-1}(z)$ , one can obtain directly a *converted measurement* of the position, which leads to a purely *linear problem* and a KF can be used. While this is convenient, it should be noted that the inverse transformation of the measurement can lead to errors that are difficult (but not impossible — see [Lerro93]) to account for.
2. By leaving the measurement in its original form, one has a *mixed coordinate filter*.

The filter was implemented in *mixed coordinates*: The measurement was in the line-of-sight angle coordinate, while the target's state was in Cartesian coordinates.

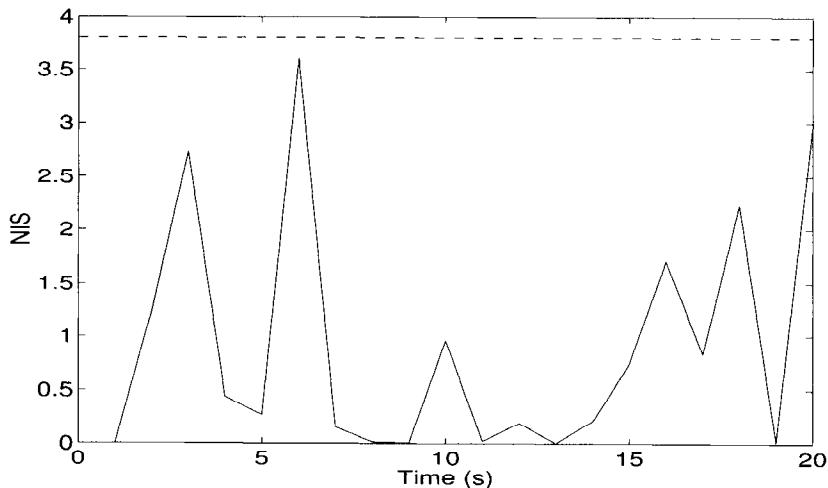
Figure 10.3.4-3 shows, for a single run, the position and velocity estimates compared to the true values. The filter appears to converge, in the sense that the estimation errors tend to decrease, but slowly.



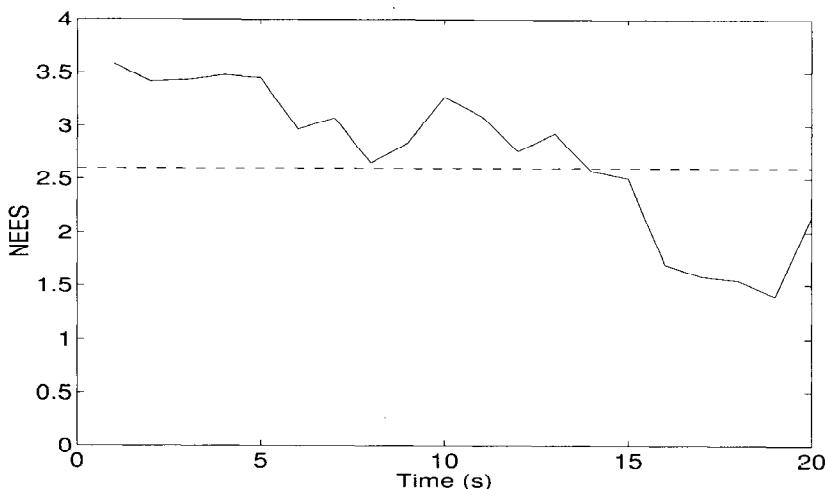
**Figure 10.3.4-3:** True and estimated position and velocity in a single run (— truth; — estimated).



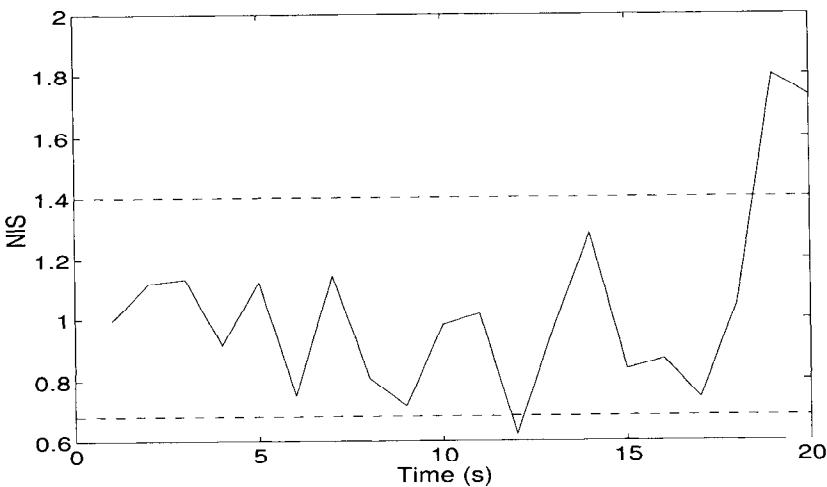
**Figure 10.3.4-4:** Normalized state estimation error squared (single run) with the one-sided 95% probability region.



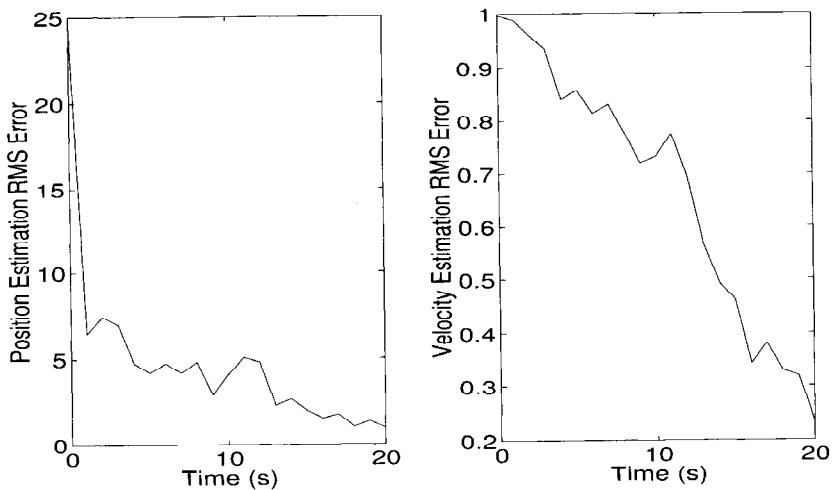
**Figure 10.3.4-5:** Normalized innovation squared (single run) with the one-sided 95% probability region.



**Figure 10.3.4-6:** Normalized state estimation error squared (50 runs) with the 97.5% probability level (upper limit of the two-sided 95% region).



**Figure 10.3.4-7:** Normalized innovation squared (50 runs) with the two-sided 95% region (2.5% and 97.5% levels).



**Figure 10.3.4-8:** Position and velocity estimation RMS error (50 runs).

The normalized state estimation error squared (NEES) shown in Fig. 10.3.4-4 for a single run exceeds the 95% level only for a short time. The normalized innovation squared shown in Fig. 10.3.4-5 is below the 95% level. Thus, from this single run, the filter appears to be consistent.

The 50-run Monte Carlo averages in Fig. 10.3.4-6 show, however, that the errors are most of the time outside the 95% probability region. This indicates that compared to the filter-calculated covariance the errors are inadmissibly large. The normalized innovation squared shown in Fig. 10.3.4-7 exhibits large deviations for the last two sampling times. Thus, this filter is *inconsistent*.

The RMS errors in position and velocity estimates are plotted in Fig. 10.3.4-8.

The conclusion from these simulations is that a *thorough examination* of a nonlinear filter is needed *via Monte Carlo runs* to find out if it is consistent.

## Remarks

The inconsistency is due to the fact that the linearization does not work well with large errors — the measurement noise in this case. Even the second-order EKF does not work well in this example.<sup>4</sup>

It can be shown that if the measurement is more accurate ( $\sqrt{r_s} \leq 1^\circ$ ), this filter becomes consistent.

### 10.3.5 The EKF — Summary

Model for the *extended Kalman filter (EKF)*:

- *Initial state* — an estimate (approximate conditional mean) available with an associated covariance.
- *System dynamic equation* — nonlinear with additive zero-mean white process noise with known covariance.
- *Measurement equation* — nonlinear with additive zero-mean white measurement noise, with known covariance.

The initial state error and the two noise sequences are assumed to be mutually independent.

The discrete-time recursive state estimation filter is obtained by using a series expansion of

- first order or
- second order

of the nonlinear dynamic and measurement equations to obtain the state and measurement predictions. These yield, respectively,

---

<sup>4</sup>Negative evidence has a way of spoiling a nice theory, or, at least, pointing to its limitations.

- the (standard, or first-order) EKF and
- the second-order EKF.

The first-order EKF requires evaluation of the Jacobians of the nonlinear functions from the dynamics and the measurement equations. The second-order EKF requires also the evaluation of the corresponding Hessians.

The first order discrete time EKF consists of the following:

1. State prediction — the nonlinear function (dynamic equation) of the previous state estimate.
2. Covariance of the state prediction — same covariance propagation equation as in the linear case except for the Jacobian of the dynamic equation (evaluated at the previous state estimate) playing the role of the transition matrix.
3. Measurement prediction — the nonlinear function (measurement equation) of the predicted state.
4. Covariance of the predicted measurement (equal to the innovation covariance) — same equation as in the linear case except for the Jacobian of the measurement equation (evaluated at the predicted state) playing the role of the measurement matrix.
5. Filter gain calculation — same as in the linear case.
6. State update — same as in the linear case.
7. Covariance of the updated state — same as in the linear case.

Note the coupling between the state estimation and covariance calculations: The latest state estimates are used in the *linearization* (Jacobian evaluations) of the nonlinear dynamic and/or measurement equations.

## 10.4 ERROR COMPENSATION IN LINEARIZED FILTERS

### 10.4.1 Some Heuristic Methods

The extended Kalman filter presented in the previous section was obtained by a Taylor series expansion up to first-order terms (which amounts to linearization) or up to second-order terms. Both these filters will, obviously, introduce errors in the equations where such an expansion is used due to the following:

1. The higher order terms that have been neglected.
2. The evaluations of the Jacobians (and Hessians) being done at estimated or predicted values of the state rather than the exact values, which are not available.

There are several ways of compensating for these errors:

1. Addition of ***artificial process noise*** or ***pseudo-noise*** for compensation of the errors in the state prediction (10.3.2-4). This can be done by using

in the covariance prediction equation (10.3.2-6) a *larger modified process noise covariance*

$$Q_m(k) = Q_p(k) + Q(k) \geq Q(k) \quad (10.4.1-1)$$

where  $Q_p \geq 0$  is the positive semidefinite pseudo-noise covariance.

2. Multiplication of the state covariance by a scalar  $\phi > 1$  at every sampling time. This amounts to letting

$$P_\phi(k+1|k) = \phi P(k+1|k) \quad (10.4.1-2)$$

and then using this “jacked up” prediction covariance matrix  $P_\phi(k+1|k)$  in the covariance update equation. The scalar  $\phi$  is called *fudge factor*.<sup>5</sup>

Alternatively, instead of (10.4.1-2), one can use a matrix

$$\Phi = \text{diag}(\sqrt{\phi_i}) \quad (10.4.1-3)$$

to obtain a selectively jacked-up state covariance

$$P^*(k+1|k) = \Phi P(k+1|k) \Phi \quad (10.4.1-4)$$

The multiplication of the state covariance by a scalar greater than unity is equivalent to the filter having a *fading memory*. At every sampling time the past data undergoes a process of *discounting* by attaching to the estimate based on it a higher covariance (lower accuracy).

### Effect of Increasing the Covariance Matrices

The increase of the state covariance will cause the filter gain to be larger, thus giving more weight to the most recent data.

Conversely, increasing the variance of the measurement noise, which will increase the innovation covariance, would lower the filter gain.

These (band-aid) techniques are completely heuristic, and they should be used only as a last resort. The flaw in them is that they implicitly assume the linearization errors are zero-mean and white.

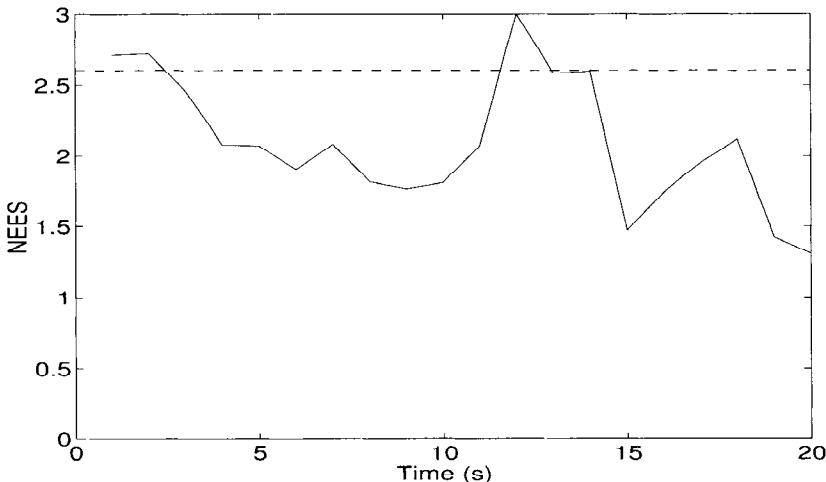
If the inconsistency stems from bias due to the nonlinearities in the system, this method will probably not accomplish much. Bias compensation along the lines of [Lerro93, Mo98] should be attempted.

### 10.4.2 An Example of Use of the Fudge Factor

The example of Subsection 10.3.4 is considered again. Since the filter did not pass the consistency tests, the covariance increase technique (10.4.1-2) is illustrated for this problem.

---

<sup>5</sup>From the Greek *fudge φακτόρ*.



**Figure 10.4.2-1:** Normalized state estimation error squared ( $\phi = 1.03$ , 50 runs) with the 97.5% level.

The fudge factor was taken as  $\phi = 1.03$ . The resulting normalized state estimation error, plotted in Fig. 10.4.2-1, shows still some points above the 97.5 percentile point. The normalized innovation in Fig. 10.4.2-2 is also outside the admissible region at the end of the observation period.

The magnitudes of the RMS errors shown in Fig. 10.4.2-3 are practically the same as in the uncompensated filter, shown earlier in Fig. 10.3.4-8.

### 10.4.3 An Example of Debiasing: Conversion from Polar to Cartesian

In a typical (two-dimensional) radar, the measured values of position are in **polar coordinates**—**range** and **azimuth** (or **bearing**). Since the motion equations of targets are naturally expressed in Cartesian coordinates, if the polar measurements can be converted to Cartesian (via a nonlinear transformation) without bias and obtain the correct covariance for the converted measurements, one can then perform the state estimation within a completely linear framework.

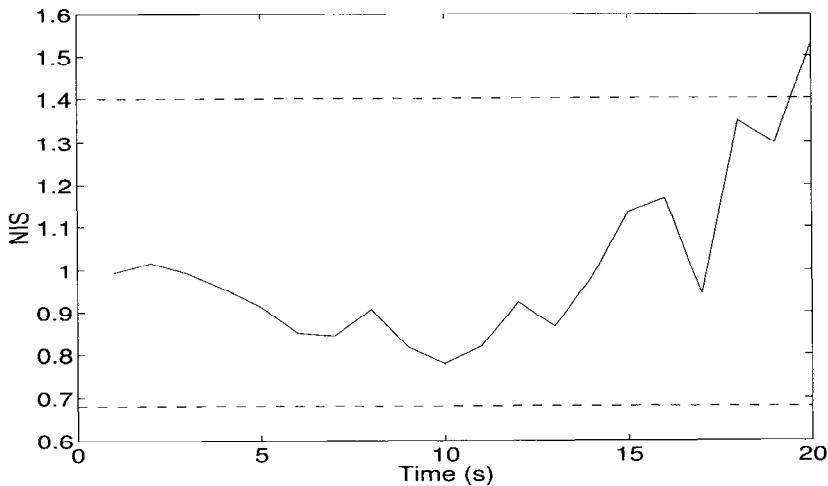
#### The Standard Conversion

The measured range and azimuth (bearing) are

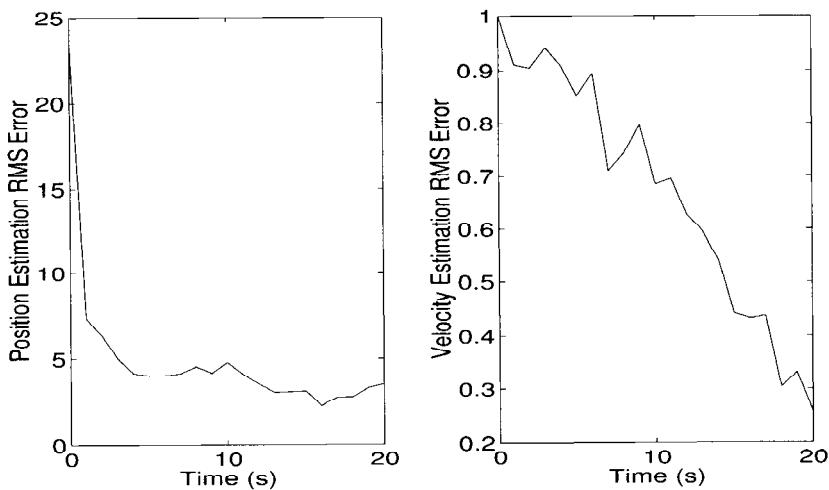
$$r_m = r + w_r \quad \theta_m = \theta + w_\theta \quad (10.4.3-1)$$

where  $r$  and  $\theta$  are the true range and azimuth, respectively, and  $w_r$  and  $w_\theta$  are independent measurement noises with moments

$$E[w_r] = 0 \quad E[w_\theta] = 0 \quad E[w_r^2] = \sigma_r^2 \quad E[w_\theta^2] = \sigma_\theta^2 \quad (10.4.3-2)$$



**Figure 10.4.2-2:** Normalized innovation squared ( $\phi = 1.03$ , 50 runs) with the two-sided 95% probability region.



**Figure 10.4.2-3:** Position and velocity estimation RMS error ( $\phi = 1.03$ , 50 runs).

with  $\sigma_r$  and  $\sigma_\theta$  the corresponding measurement error standard deviations.

The standard conversion to Cartesian is

$$x_m = r_m \cos \theta_m \quad y_m = r_m \sin \theta_m \quad (10.4.3-3)$$

with (approximate) covariance terms based on linearization

$$R_{11}^L \triangleq \text{var}(x_m) = r_m^2 \sigma_\theta^2 \sin^2 \theta_m + \sigma_r^2 \cos^2 \theta_m \quad (10.4.3-4)$$

$$R_{22}^L \triangleq \text{var}(y_m) = r_m^2 \sigma_\theta^2 \cos^2 \theta_m + \sigma_r^2 \sin^2 \theta_m \quad (10.4.3-5)$$

$$R_{12}^L \triangleq \text{cov}(x_m, y_m) = (\sigma_r^2 - r_m^2 \sigma_\theta^2) \sin \theta_m \cos \theta_m \quad (10.4.3-6)$$

### The Unbiased Conversion

Using (10.4.3-1), the independence of the noises in (10.4.3-3), and the additional assumption that they have symmetric pdf, one has

$$E[x_m] = E[r_m]E[\cos \theta_m] = r \cos \theta E[\cos w_\theta] \triangleq b_1 x \quad (10.4.3-7)$$

where  $x$  is the true coordinate. Similarly,

$$E[y_m] = b_1 y \quad (10.4.3-8)$$

Note that the bias is *multiplicative*, and is given, for Gaussian noise, by [Mo98]

$$b_1 \triangleq E[\cos w_\theta] = e^{-\sigma_\theta^2/2} \quad (10.4.3-9)$$

See also problem 10-6.

Thus the unbiased converted measurements are

$$\boxed{x_m^u = b_1^{-1} r_m \cos \theta_m \quad y_m^u = b_1^{-1} r_m \sin \theta_m} \quad (10.4.3-10)$$

### Validity Limit of the Standard Conversion

The bias in the two coordinates is

$$\mu_x = x - E[x_m] = x(1 - b_1) \quad (10.4.3-11)$$

$$\mu_y = y - E[y_m] = y(1 - b_1) \quad (10.4.3-12)$$

Thus its norm is

$$\|\mu\| = r(1 - b_1) \quad (10.4.3-13)$$

The maximum bias magnitude is, from (10.4.3-9), using a series expansion,

$$\|\mu\|_{\max} = \frac{r \sigma_\theta^2}{2} \quad (10.4.3-14)$$

Note that  $\sigma_\theta$  above is in *radians*.

This is to be compared to the minimum standard deviation of the measured position in the range and cross-range directions

$$\sqrt{\lambda_{\min}} = \min(\sigma_r, r\sigma_\theta) \quad (10.4.3-15)$$

where  $\lambda_{\min}$  denotes the minimum eigenvalue of position covariance matrix.

Define the **bias significance** as the ratio of the above two quantities

$$\beta \triangleq \frac{r\sigma_\theta^2}{2 \min(\sigma_r, r\sigma_\theta)} \quad (10.4.3-16)$$

The maximum bias that can be tolerated is, typically, 20% of the standard deviation, i.e.,

$$\beta_{\max} \leq 0.2 \quad (10.4.3-17)$$

Thus the **validity limit of the standard conversion** is

$$\frac{r\sigma_\theta^2}{\sigma_r} < 0.4 \quad (10.4.3-18)$$

$$\sigma_\theta < 0.4 \text{ rad} \approx 23^\circ \quad (10.4.3-19)$$

In practically all systems, inequality (10.4.3-19) is satisfied. In most radar systems, inequality (10.4.3-18) is satisfied, but this is not always the case in sonar systems or in long-range radar. This is a useful guideline for the limit of validity of the standard conversion.

### Covariance of the Unbiased Conversion

To obtain the covariance of the unbiased conversion, the squared error should be averaged *conditioned on the observations*. Specifically,

$$\begin{aligned} \text{var}(x_m^u) &= E\{[b_1^{-1}r_m \cos \theta_m - r \cos \theta]^2 | r_m, \theta_m\} \\ &= E\{[b_1^{-1}r_m \cos \theta_m - (r_m - w_r) \cos(\theta_m - w_\theta)]^2 | r_m, \theta_m\} \\ &= E\{b_1^{-2}r_m^2 \cos^2 \theta_m + (r_m - w_r)^2 \cos^2(\theta_m - w_\theta) \\ &\quad - 2b_1^{-1}(r_m - w_r)r_m[\cos \theta_m \cos w_\theta + \sin \theta_m \sin w_\theta] \cos \theta_m\} \\ &= b_1^{-2}r_m^2 \cos^2 \theta_m + (r_m^2 + \sigma_r^2)(1/2)E[1 + \cos 2(\theta_m - w_\theta)] \\ &\quad - 2b_1^{-1}r_m^2 \cos^2 \theta_m E[\cos w_\theta] \\ &= b_1^{-2}r_m^2 \cos^2 \theta_m + (r_m^2 + \sigma_r^2)(1/2)E[1 + \cos 2\theta_m \cos 2w_\theta \\ &\quad + \sin 2\theta_m \sin 2w_\theta] - 2r_m^2 \cos^2 \theta_m \\ &= b_1^{-2}r_m^2 \cos^2 \theta_m + (r_m^2 + \sigma_r^2)(1/2)\{1 + \cos 2\theta_m E[\cos 2w_\theta]\} \\ &\quad - 2r_m^2 \cos^2 \theta_m \\ &\stackrel{\Delta}{=} b_1^{-2}r_m^2 \cos^2 \theta_m + (r_m^2 + \sigma_r^2)(1/2)\{1 + b_2 \cos 2\theta_m\} \\ &\quad - 2r_m^2 \cos^2 \theta_m \end{aligned} \quad (10.4.3-20)$$

Note that if  $w_\theta \sim \mathcal{N}(0, \sigma_\theta^2)$ , then  $2w_\theta \sim \mathcal{N}(0, 4\sigma_\theta^2)$ , and

$$b_2 \triangleq E[\cos 2w_\theta] = e^{-2\sigma_\theta^2} = b_1^4 \quad (10.4.3-21)$$

Thus,

$$R_{11}^u \triangleq \text{var}(x_m^u) = [b_1^{-2} - 2]r_m^2 \cos^2 \theta_m + (r_m^2 + \sigma_r^2)(1/2)\{1 + b_1^4 \cos 2\theta_m\} \quad (10.4.3-22)$$

Similarly,

$$R_{22}^u \triangleq \text{var}(y_m^u) = [b_1^{-2} - 2]r_m^2 \sin^2 \theta_m + (r_m^2 + \sigma_r^2)(1/2)\{1 - b_1^4 \cos 2\theta_m\} \quad (10.4.3-23)$$

The covariance between the errors is

$$\begin{aligned} \text{cov}(x_m^u, y_m^u) &= E\{[b_1^{-1}r_m \cos \theta_m - r \cos \theta][b_1^{-1}r_m \sin \theta_m - r \sin \theta]\}|r_m, \theta_m\} \\ &= E\{[b_1^{-1}r_m \cos \theta_m - (r_m - w_r) \cos(\theta_m - w_\theta)] \\ &\quad \cdot [b_1^{-1}r_m \sin \theta_m - (r_m - w_r) \sin(\theta_m - w_\theta)]\}|r_m, \theta_m\} \\ &= E\{b_1^{-2}r_m^2 \cos \theta_m \sin \theta_m \\ &\quad + (r_m - w_r)^2 \cos(\theta_m - w_\theta) \sin(\theta_m - w_\theta) \\ &\quad - b_1^{-1}(r_m - w_r)r_m [\sin \theta_m \cos w_\theta - \cos \theta_m \sin w_\theta] \cos \theta_m \\ &\quad - b_1^{-1}(r_m - w_r)r_m [\cos \theta_m \cos w_\theta + \sin \theta_m \sin w_\theta] \sin \theta_m\} \\ &= b_1^{-2}r_m^2 \cos \theta_m \sin \theta_m + (r_m^2 + \sigma_r^2)(1/2)E[\sin 2(\theta_m - w_\theta)] \\ &\quad - b_1^{-1}r_m^2 \sin 2\theta_m E[\cos w_\theta] \\ &= b_1^{-2}r_m^2 \cos \theta_m \sin \theta_m + (r_m^2 + \sigma_r^2)(1/2) \\ &\quad \cdot E[\sin 2\theta_m \cos 2w_\theta - \cos 2\theta_m \sin 2w_\theta] - r_m^2 \sin 2\theta_m \\ &= b_1^{-2}r_m^2 \cos \theta_m \sin \theta_m + (r_m^2 + \sigma_r^2)(1/2) \sin 2\theta_m E[\cos 2w_\theta] \\ &\quad - r_m^2 \sin 2\theta_m \\ &= b_1^{-2}r_m^2 \cos \theta_m \sin \theta_m + (r_m^2 + \sigma_r^2)(1/2)b_2 \sin 2\theta_m \\ &\quad - r_m^2 \sin 2\theta_m \end{aligned} \quad (10.4.3-24)$$

which can be written as

$$R_{12}^u \triangleq \text{cov}(x_m^u, y_m^u) = [b_1^{-2}r_m^2/2 + (r_m^2 + \sigma_r^2)b_2/2 - r_m^2] \sin 2\theta_m \quad (10.4.3-25)$$

## Consistency Evaluation of the Conversions

The analysis is done by performing the statistical consistency check along the lines of Section 5.4 as follows.

For a zero-mean random vector  $\tilde{z}$  of dimension  $n_z$  and covariance  $P$  the expected value of the **Normalized Error Squared (NES)**

$$\psi = \tilde{z}' P^{-1} \tilde{z} \quad (10.4.3-26)$$

is equal to  $n_z$ . The sample mean of (10.4.3-26) from  $n$  independent realizations

$$\bar{\psi} = \frac{1}{N} \sum_{i=1}^N \tilde{z}_i' P^{-1} \tilde{z}_i \quad (10.4.3-27)$$

with  $\tilde{z}_i$  the two-dimensional vector of converted measurement errors in realization  $i$ , is used to perform the consistency test.

The mean value of the sample test statistic is 2 when there is no bias and the assumed covariance matches the actual errors. If the errors are jointly Gaussian then the distribution of (10.4.3-27) multiplied by  $N$  is chi-square with  $2N = 2000$  degrees of freedom and an acceptance region is used for the test.

The above test was performed using  $N = 1000$  converted position measurements for each accuracy considered.

The average NES for the unbiased conversion (10.4.3-10) with covariance  $R^u$  (10.4.3-23)–(10.4.3-25) evaluated at the measured position is plotted as a function of the bearing error in the top part of Fig. 10.4.3-1. The true target position is at  $10^5$  m with azimuth  $45^\circ$  and the standard deviation of range error is 50 m. The plot also indicates the chi-square 0.99 probability bounds.

The same results for the standard conversion (10.4.3-3) with the covariance  $R^L$  (10.4.3-4)–(10.4.3-6) based on linearization are shown in the bottom part of Figure 10.4.3-1.

The covariance  $R^u$  is consistent with the errors in the unbiased conversion even though the variability in the plot marginally violates the chi-square bounds since the errors are not Gaussian. The linear approximation for the standard conversion results in an *optimistic covariance*  $R^L$ , which becomes inconsistent when the bearing error exceeds  $0.8^\circ$  at this range.

Similar results are obtained at different bearing angles.

#### 10.4.4 Error Compensation in Linearized Filters — Summary

Linearization of nonlinear systems to obtain linear filters introduces unavoidable errors.

One way to account for these errors is to increase the covariances by

- addition of *pseudo-noise covariance* or
- multiplication of the state covariance by a *fudge factor* (slightly) larger than unity

so that the covariance “covers” the errors (in the sense of filter consistency).

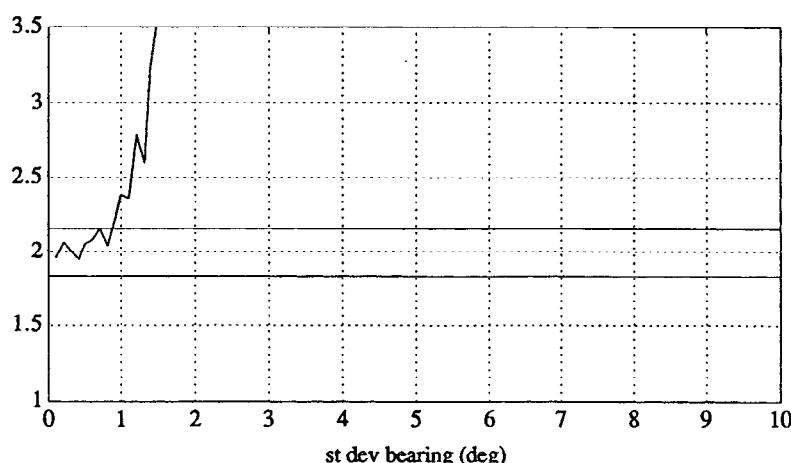
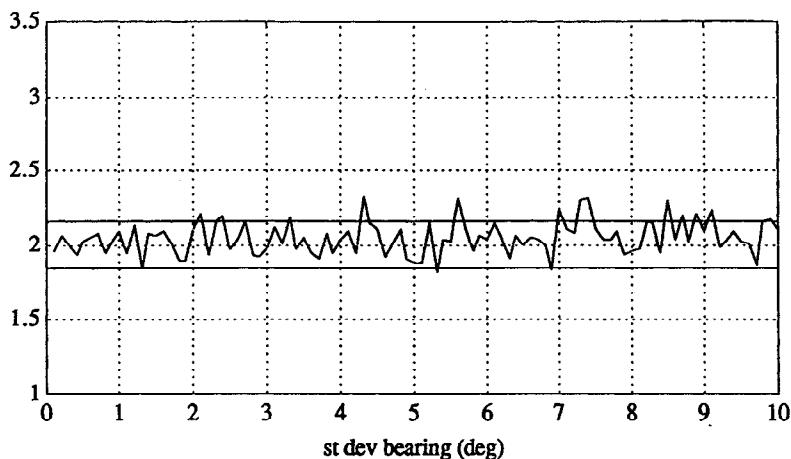
This implicitly assumes that these errors are random.

If the errors are not random, i.e., one has *bias*, then

- bias compensation, e.g., a debiasing technique as illustrated, or
- bias estimation

is needed.

Bias estimation can be accomplished by adding extra state components. In this case one has to ascertain that the augmented state is completely observable.



**Figure 10.4.3-1:** Average NES for the unbiased conversion (top) and for the standard conversion (bottom) evaluated at the measured position ( $r = 10^5$  m,  $\theta = 45^\circ$ ,  $\sigma_r = 50$  m).

## 10.5 SOME ERROR REDUCTION METHODS

### 10.5.1 Improved State Prediction

The extended Kalman filter carries out the state prediction and measurement prediction according to (10.3.2-4) and (10.3.2-7), respectively. Both these equations involve nonlinear transformations.

Applying a nonlinear transformation on an estimate introduces errors that are not completely accounted for in the prediction covariance equation. The second-order state prediction equation (10.3.2-4) does capture some of the nonlinearity effects. However, since it can be expensive to implement and there is no guarantee that this will solve the problem, a different approach is adopted in some cases.

For *continuous-time nonlinear systems* whose state evolves according to nonlinear differential equations and is observed at discrete times, one can obtain the predicted state by discretization and linearization, which amounts to a first-order numerical integration.

If this leads to unsatisfactory accuracy, then one can use a more accurate *numerical integration* on the continuous-time differential equation of the state from  $t_k$  to  $t_{k+1}$  starting at the initial condition  $\hat{x}(k|k)$  with the process noise replaced by its mean (zero).

The calculation of the state prediction covariance can be done as in (10.3.2-6), that is, relying upon linearization or series expansion if the process noise is specified in discrete time, or by numerical integration of (9.3.1-8) with the system matrix replaced by the Jacobian of the nonlinear dynamic equation (see problem 10-4).

The continuous-discrete filter for nonlinear systems, which relies on propagation of the state's pdf via the Fokker-Planck equation between the measurement times, is discussed in Section 10.7.

### 10.5.2 The Iterated Extended Kalman Filter

A modified state updating approach can be obtained by an iterative procedure as follows.

The measurement prediction based on expansion up to first order is

$$\hat{z}(k+1|k) = h[k+1, \hat{x}(k+1|k)] \quad (10.5.2-1)$$

that is, the predicted state  $\hat{x}(k+1|k)$  is used for  $x(k+1)$ . Since the state prediction errors might already be significant, when compounded with additional errors due to the measurement nonlinearity, this can lead to undesirable errors.

One approach to alleviate this situation is to compute the updated state not as an approximate conditional mean — that is, a linear combination of the prediction and the innovation (5.2.3-12) — but as a maximum a posteriori

(MAP) estimate. An *approximate MAP estimate* can be obtained by an iteration that amounts to ***relinearization of the measurement equation***.

The conditional pdf of  $x(k+1)$  given  $Z^{k+1}$  can be written, assuming all the pertinent random variables to be Gaussian, as

$$\begin{aligned} p[x(k+1)|Z^{k+1}] &= p[x(k+1)|z(k+1), Z^k] \\ &= \frac{1}{c} p[z(k+1)|x(k+1)] p[x(k+1)|Z^k] \\ &= \frac{1}{c} \mathcal{N}[z(k+1); h[k+1, x(k+1)], R(k+1)] \\ &\quad \cdot \mathcal{N}[x(k+1); \hat{x}(k+1|k), P(k+1|k)] \end{aligned} \quad (10.5.2-2)$$

Maximizing the above with respect to  $x(k+1)$  is equivalent to minimizing

$$\begin{aligned} J[x(k+1)] &= \frac{1}{2} \{z(k+1) - h[k+1, x(k+1)]\}' R(k+1)^{-1} \\ &\quad \cdot \{z(k+1) - h[k+1, x(k+1)]\} \\ &\quad + \frac{1}{2} [x(k+1) - \hat{x}(k+1|k)]' P(k+1|k)^{-1} \\ &\quad \cdot [x(k+1) - \hat{x}(k+1|k)] \end{aligned} \quad (10.5.2-3)$$

The iterative minimization of (10.5.2-3), say, using a Newton-Raphson algorithm, will yield an *approximate MAP estimate* of  $x(k+1)$ . This is done by expanding  $J$  in a Taylor series up to second order about the  $i$ th iterated value of the estimate of  $x(k+1)$ , denoted (without time argument) as  $x^i$ ,

$$J = J^i + J_x^i(x - x^i) + \frac{1}{2}(x - x^i)' J_{xx}^i(x - x^i) \quad (10.5.2-4)$$

where, using abbreviated notation,

$$J^i = J \Big|_{x=x^i} \quad (10.5.2-5)$$

and

$$J_x^i = \nabla_x J \Big|_{x=x^i} \quad (10.5.2-6)$$

$$J_{xx}^i = \nabla_x \nabla_x' J \Big|_{x=x^i} \quad (10.5.2-7)$$

are the gradient and Hessian of  $J$  with respect to  $x(k+1)$ .

Setting the gradient of (10.5.2-4) with respect to  $x$  to zero yields the next value of  $x$  in the iteration to minimize (10.5.2-3) as

$$x^{i+1} = x^i - (J_{xx}^i)^{-1} J_x^i \quad (10.5.2-8)$$

The gradient of  $J$  is, using now the full notation,

$$\begin{aligned} J_x^i &= -h_x^i[k+1, \hat{x}^i(k+1|k+1)]' R(k+1)^{-1} \\ &\quad \cdot \{z(k+1) - h[k+1, \hat{x}^i(k+1|k+1)]\} \\ &\quad + P(k+1|k)^{-1} [\hat{x}^i(k+1|k+1) - \hat{x}(k+1|k)] \end{aligned} \quad (10.5.2-9)$$

The Hessian of  $J$ , retaining only up to the first derivative of  $h$ , is

$$\begin{aligned} J_{xx}^i &= h_x^i[k+1, \hat{x}^i(k+1|k+1)]' R(k+1)^{-1} h_x[k+1, \hat{x}^i(k+1|k+1)] \\ &\quad + P(k+1|k)^{-1} \\ &= H^i(k+1)' R(k+1)^{-1} H^i(k+1) + P(k+1|k)^{-1} \end{aligned} \quad (10.5.2-10)$$

where

$$\boxed{H^i(k+1) \triangleq h_x[k+1, \hat{x}^i(k+1|k+1)]} \quad (10.5.2-11)$$

is the Jacobian of the *relinearized measurement equation*.

Using the matrix inversion lemma (1.3.3-11), one has

$$\begin{aligned} (J_{xx}^i)^{-1} &= P(k+1|k) - P(k+1|k) H^i(k+1)' \\ &\quad \cdot [H^i(k+1) P(k+1|k) H^i(k+1)' + R(k+1)]^{-1} \\ &\quad \cdot H^i(k+1) P(k+1|k) \\ &\triangleq P^i(k+1|k+1) \end{aligned} \quad (10.5.2-12)$$

Substituting (10.5.2-9) and (10.5.2-12) into (10.5.2-8) yields

$$\boxed{\begin{aligned} \hat{x}^{i+1}(k+1|k+1) &= \hat{x}^i(k+1|k+1) + P^i(k+1|k+1) H^i(k+1)' \\ &\quad \cdot R(k+1)^{-1} \{ z(k+1) - h[k+1, \hat{x}^i(k+1|k+1)] \} \\ &\quad - P^i(k+1|k+1) P(k+1|k)^{-1} \\ &\quad \cdot [\hat{x}^i(k+1|k+1) - \hat{x}(k+1|k)] \end{aligned}} \quad (10.5.2-13)$$

which is the **iterated extended Kalman filter**.

Starting the iteration for  $i = 0$  with

$$\hat{x}^0(k+1|k+1) \triangleq \hat{x}(k+1|k) \quad (10.5.2-14)$$

causes the last term in (10.5.2-13) to be zero and yields after the first iteration  $\hat{x}^1(k+1|k+1)$ , that is, the same as the first-order (noniterated) EKF.

The covariance associated with  $\hat{x}^i(k+1|k+1)$  is, from (10.5.2-12), given by

$$\boxed{\begin{aligned} P^i(k+1|k+1) &= P(k+1|k) - P(k+1|k) H^i(k+1)' \\ &\quad \cdot [H^i(k+1) P(k+1|k) H^i(k+1)' + R(k+1)]^{-1} \\ &\quad \cdot H^i(k+1) P(k+1|k) \end{aligned}} \quad (10.5.2-15)$$

## Overview of the Iteration Sequence

For  $i = 0, N - 1$

$$(10.5.2-11)$$

$$(10.5.2-15)$$

$$(10.5.2-13)$$

For  $i = N$

$$(10.5.2-15)$$

with  $N$  decided either a priori or based on a convergence criterion.

### 10.5.3 Some Error Reduction Methods — Summary

In systems with continuous-time nonlinear dynamic equations the state prediction can be done by *numerical integration* of the dynamic equation.

This is more accurate than discretizing the continuous-time equation and using a truncated series expansion.

Another way to reduce the effects of linearization errors in the EKF is to *relinearize the measurement equation* around the *updated state* rather than relying only on the *predicted state*.

This is equivalent to a MAP estimation of the state under the (obviously approximate) Gaussian assumption carried out via a Newton-Raphson search.

It results in a set of coupled iterations for

- The updated state
- The updated state covariance
- The Jacobian of the measurement equation

## 10.6 MAXIMUM A POSTERIORI TRAJECTORY ESTIMATION VIA DYNAMIC PROGRAMMING

### 10.6.1 The Approach

This approach finds the sequence of MAP estimates

$$\begin{aligned}\hat{x}^{k|k} &\triangleq \{\hat{x}(0|k), \dots, \hat{x}(k|k)\} \\ &= \arg \max_{x(0), \dots, x(k)} p[x(0), \dots, x(k)|Z^k]\end{aligned}\quad (10.6.1-1)$$

These estimates, which are actually smoothed values, constitute the *modal trajectory*. This is the maximum — the *mode* — of the joint pdf of the state trajectory, that is, the most probable sequence of state values up to the current time.

Equation (10.6.1-1) can be rewritten equivalently as

$$p[\hat{x}^{k|k}|Z^k] = \max_{X^k} p[X^k|Z^k] \quad (10.6.1-2)$$

where

$$X^k \triangleq \{x(j)\}_{j=0}^k \quad (10.6.1-3)$$

is the set of variables w.r.t. which the maximization is to be performed.

The technique of *dynamic programming* is used. This optimization technique replaces the simultaneous maximization (10.6.1-2) by a *sequential (multistage) maximization*. The last maximization is

$$p[\hat{x}^{k|k}|Z^k] = \max_{x(k)} \max_{X^{k-1}} p[X^k|Z^k] \quad (10.6.1-4)$$

This technique can be used if the function to be maximized can be decomposed (separated) at each stage into

- A sum or product of two functions
- Where the first function depends only on the first variable

Then one can obtain a sequential maximization.

The function to be maximized recursively is

$$I[x(k), k] = \max_{X^{k-1}} p[X^k | Z^k] \quad (10.6.1-5)$$

from which (10.6.1-4) follows by one more maximization.

## 10.6.2 The Dynamic Programming for Trajectory Estimation

Using Bayes' formula and eliminating irrelevant variables in the conditioning, (10.6.1-5) can be rewritten as

$$\begin{aligned} I[x(j+1), j+1] &= \max_{X^j} p[X^{j+1} | Z^{j+1}] = \max_{X^j} p[x(j+1), X^j | z(j+1), Z^j] \\ &= \max_{X^j} \frac{p[z(j+1) | x(j+1), X^j, Z^j] p[x(j+1), X^j | Z^j]}{p[z(j+1) | Z^j]} \\ &= \frac{1}{c_j} \max_{X^j} \{p[z(j+1) | x(j+1)] p[x(j+1) | X^j, Z^j] \\ &\quad \cdot p[X^j | Z^j]\} \\ &= \frac{1}{c_j} \max_{x(j)} \max_{X^{j-1}} p[z(j+1) | x(j+1)] p[x(j+1) | x(j)] \\ &\quad \cdot p[X^j | Z^j] \\ &= \frac{1}{c_j} \max_{x(j)} \{p[z(j+1) | x(j+1)] p[x(j+1) | x(j)] \\ &\quad \cdot \max_{X^{j-1}} p[X^j | Z^j]\} \end{aligned} \quad (10.6.2-1)$$

where in the last line advantage has been taken of the “separability” property mentioned earlier. The term  $c_j$  is the normalization constant; it does not affect the maximization and can be dropped.

Combining (10.6.1-5) and (10.6.2-1) yields the (forward) functional recursion

$$I^*[x(j+1), j+1] = \max_{x(j)} \{p[z(j+1) | x(j+1)] p[x(j+1) | x(j)] I^*[x(j), j]\} \quad (10.6.2-2)$$

where  $I^*(j+1)$  is proportional to  $I(j+1)$ , with the proportionality factor consisting of normalization constants that are irrelevant to the estimation. This is the desired iterative relationship (multistage maximization) that yields (10.6.1-5).

The resulting argument from the maximization (10.6.2-2) is

$$\hat{x}(j|j+1) = x[j, x(j+1), Z^j] \quad j = 0, \dots, k-1 \quad (10.6.2-3)$$

that is, it is not yet the modal trajectory.

To obtain the modal trajectory, use is made of (10.6.1-4) with the final result of (10.6.2-2), which yields

$$\hat{x}(k|k) = \arg \max_{x(k)} I^*[x(k), k] \quad (10.6.2-4)$$

This is then used in (10.6.2-3), which is a backward recursion, as follows

$$\hat{x}(j|k) = x[j, \hat{x}(j+1|k), Z^j] \quad j = k-1, \dots, 0 \quad (10.6.2-5)$$

Note that the above is equivalent to a **MAP smoothing technique**.

In the communication theory literature a similar approach is known as the **Viterbi algorithm**.

## Remarks

This approach is very general, in the sense that it does not require any particular form of the state or measurement equations. The state transition equation is summarized by the conditional pdf

$$p[x(j+1)|x(j)] \quad (10.6.2-6)$$

while the measurement equation is summarized by

$$p[z(j+1)|x(j+1)] \quad (10.6.2-7)$$

This technique has been applied in [Barniv85] and also in Chapter 4 of [Bar-Shalom98a] to a problem where (10.6.2-6) describes the motion of a target and (10.6.2-7) models the observed intensities (from a target or background) in each resolution cell of an electro-optical sensor. The resulting algorithm, which can detect target trajectories in a very low SNR environment and estimate their state, carries out what amounts to a **multiscan (multiframe) signal processing**, also known as **track before detect**.

The practical implementation of this algorithm requires discretization of the state space as well as the measurement space.

## 10.7 Nonlinear Continuous-Discrete Filter

### 10.7.1 The Model

In Section 9.3 the continuous-discrete filter was discussed for linear systems. Here nonlinear systems will be considered, with a scalar state for simplicity. The system state is assumed to evolve in continuous time according to

$$\dot{x}(t) = f[t, x(t)] + \tilde{v}(t) \quad (10.7.1-1)$$

where  $\tilde{v}(t)$  is a zero-mean continuous-time white process noise with intensity  $\tilde{q}(t)$  and the measurements are obtained at discrete times

$$z(t_k) = h[t_k, x(t_k)] + w(t_k) \quad (10.7.1-2)$$

where  $w(t_k)$  is a zero-mean white sequence. The state  $x$  in Eq. (10.7.1-1) is called a *diffusion process*; if the process noise has a nonzero mean (or there is a known additive input to the system), then the diffusion process also has a *drift*.

The initial information is the pdf of the initial state. The process noise, the measurement noise and the initial state are independent of each other.

As in Section 10.2, the optimal estimation consists of calculating the conditional pdf of the state given the measurements. The *optimal nonlinear continuous-discrete filter* consists of the following:

- Propagation of the state's pdf between the measurement instants
- Use of Bayes' formula to update the state pdf at the measurement instants

The propagation of the state's pdf between the measurement instants accounting exactly (i.e., without approximations, for a change) for (10.7.1-1) will be shown to be governed by a partial differential equation. This "open-loop" propagation equation is known as the *forward diffusion equation* or the *Fokker-Planck equation* or *Kolmogorov forward equation*.

### 10.7.2 The Fokker-Planck Equation

Denote the moment generating function of the scalar random variable  $x$  given  $y$  as

$$M_{x|y}(s) = E\{e^{sx}|y\} = \int_{-\infty}^{\infty} e^{sx} p(x|y) dx = 1 + \sum_{i=1}^{\infty} \frac{s^i}{i!} m_{x|y}^i \quad (10.7.2-1)$$

where

$$m_{x|y}^i \triangleq E[x^i|y] \quad (10.7.2-2)$$

Define the increment in the state

$$\Delta x(t) \triangleq x(t) - x(t - \Delta t) \quad (10.7.2-3)$$

Then its moment generating function conditioned on  $x(t - \Delta t)$  is

$$M_{\Delta x(t)|x(t-\Delta t)}(s) = \int_{-\infty}^{\infty} e^{s[x(t)-x(t-\Delta t)]} p[x(t)|x(t-\Delta t)] dx(t) \quad (10.7.2-4)$$

and, using the inverse Fourier transformation, one has

$$p[x(t)|x(t-\Delta t)] = \frac{1}{2\pi j} \int_{-j\infty}^{j\infty} e^{-s[x(t)-x(t-\Delta t)]} M_{\Delta x(t)|x(t-\Delta t)}(s) ds \quad (10.7.2-5)$$

Consider the identity

$$p[x(t)] = \int p[x(t)|x(t-\Delta t)] p[x(t-\Delta t)] dx(t-\Delta t) \quad (10.7.2-6)$$

Using (10.7.2-5) in the above yields

$$p[x(t)] = \frac{1}{2\pi j} \int_{-\infty}^{\infty} \int_{-j\infty}^{j\infty} e^{-s\Delta x(t)} M_{\Delta x(t)|x(t-\Delta t)}(s) p[x(t-\Delta t)] dx(t-\Delta t) ds \quad (10.7.2-7)$$

Similarly to (10.7.2-1), one has

$$M_{\Delta x(t)|x(t-\Delta t)}(s) = 1 + \sum_{i=1}^{\infty} \frac{s^i}{i!} m_{\Delta x(t)|x(t-\Delta t)}^i \quad (10.7.2-8)$$

Using the above in (10.7.2-7) yields

$$\begin{aligned} p[x(t)] &= \frac{1}{2\pi j} \int_{-\infty}^{\infty} \int_{-j\infty}^{j\infty} e^{-s\Delta x(t)} \left[ 1 + \sum_{i=1}^{\infty} \frac{s^i}{i!} m_{\Delta x(t)|x(t-\Delta t)}^i \right] \\ &\quad \cdot p[x(t-\Delta t)] dx(t-\Delta t) ds \end{aligned} \quad (10.7.2-9)$$

Consider the identities

$$\frac{1}{2\pi j} \int_{-j\infty}^{j\infty} s^i e^{-s\Delta x(t)} ds = (-1)^i \frac{\partial^i \delta[\Delta x(t)]}{[\partial \Delta x(t)]^i} = (-1)^i \frac{\partial^i \delta[\Delta x(t)]}{[\partial x(t)]^i} \quad i = 0, 1, \dots \quad (10.7.2-10)$$

where  $\delta[\cdot]$  is the Dirac delta (impulse) function.

Using the above in (10.7.2-9) and indicating by subscript to which random variable a pdf corresponds, one has

$$\begin{aligned} p_{x(t)}[x(t)] &= \int_{-\infty}^{\infty} \delta[\Delta x(t)] p_{x(t-\Delta t)}[x(t-\Delta t)] dx(t-\Delta t) \\ &+ \sum_{i=1}^{\infty} \frac{(-1)^i}{i!} \int_{-\infty}^{\infty} \frac{\partial^i \delta[\Delta x(t)]}{[\partial x(t)]^i} m_{\Delta x(t)|x(t-\Delta t)}^i p_{x(t-\Delta t)}[x(t-\Delta t)] dx(t-\Delta t) \end{aligned} \quad (10.7.2-11)$$

Since the first term on the right-hand side above is  $p_{x(t-\Delta t)}[x(t)]$ , one has

$$p_{x(t)}[x(t)] - p_{x(t-\Delta t)}[x(t)] = \sum_{i=1}^{\infty} \frac{(-1)^i}{i!} \frac{\partial^i}{[\partial x(t)]^i} \{m_{\Delta x(t)|x(t-\Delta t)}^i p_{x(t-\Delta t)}[x(t)]\} \quad (10.7.2-12)$$

This is the variation in time (to  $t$  from  $t - \Delta t$ ) of the pdf of  $x$  evaluated at the same location  $x(t)$ . Thus,

$$\frac{\partial}{\partial t} p_{x(t)}[x(t)] = \sum_{i=1}^{\infty} \frac{(-1)^i}{i!} \frac{\partial^i}{[\partial x(t)]^i} \left\{ \lim_{\Delta t \rightarrow 0} \left[ \frac{m_{\Delta x(t)|x(t-\Delta t)}^i}{\Delta t} p_{x(t-\Delta t)}[x(t)] \right] \right\} \quad (10.7.2-13)$$

is the partial differential equation of the propagation of the pdf of the state for a general scalar Markov process.

Note that

$$\lim_{\Delta t \rightarrow 0} \frac{m_{\Delta x(t)|x(t-\Delta t)}^i}{\Delta t} = \lim_{\Delta t \rightarrow 0} \frac{E[\Delta x(t)]^i}{\Delta t} \quad (10.7.2-14)$$

For the dynamic equation (10.7.1-1) one has

$$dx(t) = f[t, x(t)] dt + \tilde{v}(t) dt = f[t, x(t)] dt + d\mathbf{w}(t) \quad (10.7.2-15)$$

where

$$E\tilde{v}(t) = 0 \quad E[\tilde{v}(t)\tilde{v}(\tau)] = \tilde{q}(t)\delta(t - \tau) \quad (10.7.2-16)$$

and, consequently,

$$Ed\mathbf{w}(t) = 0 \quad E[d\mathbf{w}(t)d\mathbf{w}(t)] = \tilde{q}(t) dt \quad (10.7.2-17)$$

where  $\mathbf{w}$  denotes a Wiener process.

Therefore,

$$E[\Delta x(t)|x(t-\Delta t)] = m_{\Delta x(t)|x(t-\Delta t)}^1 = f[t, x(t)]\Delta t \quad (10.7.2-18)$$

$$E[\Delta x(t)^2|x(t-\Delta t)] = m_{\Delta x(t)|x(t-\Delta t)}^2 = \tilde{q}(t)\Delta t \quad (10.7.2-19)$$

$$E[\Delta x(t)^3|x(t-\Delta t)] = m_{\Delta x(t)|x(t-\Delta t)}^3 = \mathcal{O}[(\Delta t)^{3/2}] \quad (10.7.2-20)$$

where  $\mathcal{O}$  denotes “of the order of.”

Finally, using these in (10.7.2-13) one has only two terms left on the right-hand side, namely,

$$\frac{\partial}{\partial t} p[x(t)] = -\frac{\partial}{\partial x(t)} \{f[t, x(t)] p[x(t)]\} + \frac{1}{2} \tilde{q}(t) \frac{\partial^2}{[\partial x(t)]^2} \{p[x(t)]\} \quad (10.7.2-21)$$

where the subscripts from the pdf have been dropped because there is no more ambiguity. The above is the **Fokker-Planck (F-P) equation** or **Kolmogorov forward equation**.

In general, this equation has to be solved numerically.

### 10.7.3 Example

Consider the case of a linear system

$$\dot{x}(t) = a x(t) + \tilde{v}(t) \quad (10.7.3-1)$$

where

$$E\tilde{v}(t) = 0 \quad E[\tilde{v}(t)\tilde{v}(\tau)] = \tilde{q}(t) \delta(t - \tau) \quad (10.7.3-2)$$

The Fokker-Planck equation for this system is

$$\frac{\partial}{\partial t} p[x(t)] = -a \frac{\partial}{\partial x(t)} \{x(t) p[x(t)]\} + \frac{1}{2} \tilde{q}(t) \frac{\partial^2}{\partial x(t)^2} \{p[x(t)]\} \quad (10.7.3-3)$$

Assume a Gaussian pdf as the solution

$$p[x(t)] = \frac{1}{\sqrt{2\pi} V_x(t)^{1/2}} e^{-\frac{[x(t)-\bar{x}(t)]^2}{2V_x(t)}} \quad (10.7.3-4)$$

Note that in the above the functions of time are  $\bar{x}(t)$  and  $V_x(t)$  while  $x(t)$  is the value at which the pdf is evaluated.

The F-P equation becomes, after canceling  $\sqrt{2\pi}$  and the exponential,

$$\begin{aligned} & -\frac{1}{2} V_x(t)^{-3/2} \frac{dV_x(t)}{dt} + V_x(t)^{-3/2} [x(t) - \bar{x}(t)] \frac{d\bar{x}(t)}{dt} + \frac{1}{2} V_x(t)^{-5/2} \\ & \cdot [x(t) - \bar{x}(t)]^2 \frac{dV_x(t)}{dt} \\ & = -a V_x(t)^{-1/2} + ax(t)[x(t) - \bar{x}(t)]V_x(t)^{-3/2} \\ & + \frac{1}{2} \tilde{q} \{-V_x(t)^{-3/2} + V_x(t)^{-5/2} [x(t) - \bar{x}(t)]^2\} \end{aligned} \quad (10.7.3-5)$$

The above has to hold  $\forall x(t)$ . Equating the terms independent of  $x(t)$  yields

$$\frac{dV_x(t)}{dt} = 2a V_x(t) + \tilde{q} \quad (10.7.3-6)$$

which is the well-known open-loop variance propagation equation (4.2.4-7) or (9.2.3-12).

Equating in (10.7.3-5) the terms with  $x(t)$  and canceling  $[x(t) - \bar{x}(t)]V_x(t)^{-3/2}$  yields

$$\frac{d\bar{x}(t)}{dt} + \frac{1}{2} \frac{x(t) - \bar{x}(t)}{V_x(t)} \frac{dV_x(t)}{dt} = ax(t) + \tilde{q} \frac{x(t) - \bar{x}(t)}{V_x(t)} \quad (10.7.3-7)$$

Substituting (10.7.3-6) in the above, one has

$$\frac{d\bar{x}(t)}{dt} + \frac{1}{2} \frac{x(t) - \bar{x}(t)}{V_x(t)} [2a V_x(t) + \tilde{q}] = ax(t) + \frac{1}{2} \tilde{q} \frac{x(t) - \bar{x}(t)}{V_x(t)} \quad (10.7.3-8)$$

which, after cancellations, yields the propagation equation of the mean of the state

$$\frac{d\bar{x}(t)}{dt} = a\bar{x}(t) \quad (10.7.3-9)$$

The above derivation also proves the preservation of the Gaussianness.

## 10.8 NOTES, PROBLEMS AND A PROJECT

### 10.8.1 Bibliographical Notes

The discrete-time nonlinear estimation for Markov processes is a straightforward consequence of Bayes' formula. More on this topic can be found in, for example, [Sage71]. The concept of information state has been discussed in [Striebel65]. The application of numerical techniques for optimal estimation with update of the pdf is illustrated for some problems in [Kramer87, Kramer88, Jaarsma91]. Recent advances in nonlinear filtering are described in [Daum88, Daum94, Daum95, Schmidt93].

Historical notes on the extended Kalman filter can be found in [Sorenson85]. The second order EKF derivation presented in Section 10.3 is based on [Athans68].

The issues of error compensation discussed in Section 10.4 are treated in numerous places, see, for example [Jazwinski70]. Bias estimation and compensation has been discussed in [Friedland69]. The mixed coordinate EKF was discussed in [Mchra71]. The additive debiasing for the polar-to-Cartesian transformation is presented in [Lerro93], while the exact multiplicative debiasing is from [Mo98].

The iterated EKF described in Section 10.5 has been used for many years, see, for example, [Wishner70]. The application of this to ballistic missile tracking has been reported in [Chang84]. Other techniques for improving the performance of the EKF in the presence of significant nonlinearities were presented in [Tenney77b, Weiss80, Song85]. Issues related to bearings-only tracking have been discussed in [Nardone81, Aidala83]. The use of two angular observations to track a target moving in a three-dimensional space has been treated in [Stallard87]. A comparison of Cartesian and polar coordinates in terms of the conditioning number of the EKF-calculated covariance can be found in [Cortina91]. A Modified Gain EKF was investigated in [Song85, Galkowski91].

The use of dynamic programming for maximum a posteriori trajectory estimation, discussed in Section 10.6, was proposed in [Larson66]. This technique was used in [Barniv85] and Chapter 4 of [Bar-Shalom98a] in a "track-before-detect" approach for extracting trajectories from a low SNR optical sensor. For more details on dynamic programming, see, e.g., [Bertsekas92]. Direct implementation of the functional recursion for the update of the state's conditional pdf was investigated in [Kramer87, Bethel87].

A recent development is the use of Markov chain Monte Carlo (MCMC) methods or sequential Monte Carlo (SMC) methods for nonlinear filtering. See, e.g., [Gordon93, Doucet00, Doucet01] for more details.

Discussion on the CRLB for nonlinear estimation can be found in [Kerr89b, Tichavsky98].

### 10.8.2 Problems

**10-1 Signal interpolation for measurement generation.** Consider three noisy signal intensity observations

$$z_i = y(x_i) + w_i \quad i = 1, 2, 3$$

of the unknown function  $y(x)$  at  $x_1 < x_2 < x_3$ . We want to find, using these observa-

tions, the maximum of  $y(x)$  and its accuracy. Assuming that  $z_1 < z_2 > z_3$ , the unknown function  $y(x)$  can be approximated locally by a parabola.

1. Using a parabolic interpolation, find

$$\hat{x}_m = f(x_1, x_2, x_3, z_1, z_2, z_3)$$

the estimate of the location of the maximum of  $y(x)$ .

2. Denoting the vector of noises as  $w \triangleq [w_1 \ w_2 \ w_3]'$  and assuming

$$E[w] = 0 \quad E[ww'] = P$$

find, using a first-order series expansion, the variance associated with the estimate  $\hat{x}_m$ .

- (iii) Calculate the location of the maximum and its variance for

$$x_1 = n - 1 \quad x_2 = n \quad x_3 = n + 1$$

$$z_1 = 0.6 \quad z_2 = 0.8 \quad z_3 = 0.6 \quad E[w_i w_j] = 0.01 \delta_{ij}$$

- 10-2 Standard deviation improvement versus variance improvement.** Assume that the variance of the estimate of a variable in a certain baseline method is equal to 1. A more sophisticated method yields a smaller variance  $1 - \alpha$  (i.e., its improvement is  $(100\alpha)\%$ ). Find the improvement in % in the corresponding standard deviation, assuming  $\alpha < 0.2$ .

- 10-3 Coordinate transformation from polar to Cartesian.** Let

$$\rho = \sqrt{x_1^2 + x_2^2} \quad \theta = \tan^{-1} \frac{x_2}{x_1}$$

Consider the measurements

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \rho + w_1 \\ \theta + w_2 \end{bmatrix} = h(x_1, x_2) + w$$

with  $w \sim \mathcal{N}(0, R)$ ,  $R = \text{diag}(\sigma_\rho^2, \sigma_\theta^2)$ .

1. Find the transformation into Cartesian coordinates

$$y = h^{-1}(z) = [x_1 \ x_2]' + w_C$$

2. Find the covariance matrix  $R_C$  of the Cartesian converted measurement noise  $w_C$  using linearization at  $\rho_0 = 10^5$ ,  $\theta_0 = 45^\circ$ .
3. Simulate  $N = 100$  realizations of  $w$  with  $R = \text{diag}[100^2, (0.5^\circ)^2]$ . Calculate the corresponding polar measurements  $z$  at  $\rho = \rho_0$ ,  $\theta = \theta_0$  and obtain the resulting Cartesian positions with the solution of item 1. Find the average of  $y$ .
4. Analyze the results whether the errors  $w_C$  are zero mean.
5. Analyze whether the covariance matrix calculated as in item 2 can be accepted as correct. Hint: Use the average of  $w_C' R_C^{-1} w_C$  as the statistic for the covariance test.

**10-4 Continuous-time EKF.** Given the continuous time nonlinear dynamic system

$$\dot{x}(t) = \phi[t, x(t)] + \tilde{v}(t)$$

with measurement

$$z(t) = \eta[t, x(t)] + \tilde{w}(t)$$

where the noises are mutually uncorrelated with autocorrelations

$$E[\tilde{v}(t)\tilde{v}(\tau)'] = \tilde{q}(t)\delta(t - \tau) \quad E[\tilde{w}(t)\tilde{w}(\tau)'] = \tilde{R}(t)\delta(t - \tau)$$

and uncorrelated from the initial state, whose estimate and covariance are  $\hat{x}(0)$  and  $P(0)$ , respectively. Find the continuous time extended Kalman-Bucy filter using linearization for the above system.

**10-5 Angular rate estimation.** Consider the estimate  $\hat{x}(t)$  of the state (4.2.2-16) of an object at time  $t$ . Assume that in the interval  $(t, t + T]$  it moves with a constant speed and constant angular rate  $\Omega$ .

1. Obtain the extrapolated state  $\hat{x}(t + T) = e^{AT}\hat{x}(t)$  where the transition matrix for this motion is given in (4.2.2-19).
2. Given the covariance matrix  $P(t)$  associated with  $\hat{x}(t)$  find the covariance associated with  $\hat{x}(t + T)$ .
3. A position measurement

$$z(t + T) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x(t + T) + w(t + T)$$

with  $w \sim \mathcal{N}(0, \sigma^2 I)$  is made. Write the likelihood function

$$p[z(t + T)|\hat{x}(t), \Omega]$$

assuming  $\hat{x}(t) \sim \mathcal{N}[0, P(t)]$ . Indicate the explicit expressions of  $\hat{x}(t + T)$  and its associated covariance  $S(t + T)$ .

4. How can one obtain  $\hat{\Omega}^{\text{ML}}$  from the above?
5. Assume, for simplicity, that  $\Omega T \ll 1$  and  $S(t + T) = cI$  and obtain the LS estimate  $\hat{\Omega}$  and its variance.

**10-6 Polar-to-Cartesian conversion bias.** Prove (10.4.3-7)–(10.4.3-9).**10-7 Speed and course estimates.** Given the estimate and covariance matrix of the state vector  $x = [x_1 \ x_2 \ x_3 \ x_4]' = [\xi \ \eta \ \dot{\xi} \ \dot{\eta}]'$  of a point moving in a plane as  $\hat{x} = [\hat{x}_1 \ \hat{x}_2 \ \hat{x}_3 \ \hat{x}_4]'$  and  $P = [P_{ij}]$ , respectively. Assume

$$|\hat{x}_i|/\sqrt{P_{ii}} \ll 1 \quad \forall i$$

1. Give the expression of the estimate of its speed  $v$  (the magnitude of the velocity vector).

2. Find the variance of the error for this speed estimate.
3. Give the expression of the estimate of the course angle  $\theta$ , defined *clockwise* from North (the  $\eta$  coordinate).
4. Find the variance of the error for this course estimate.
5. Give the expression of the RMS course error in degrees.
6. Evaluate items 1–4 above for

$$\hat{x} = [100 \text{ m} \ 100 \text{ m} \ 10 \text{ m/s} \ 10 \text{ m/s}]'$$

$$P = \begin{bmatrix} 100 & 10 & 2 & 1 \\ 10 & 100 & 1 & 2 \\ 2 & 1 & 1 & 0.5 \\ 1 & 2 & 0.5 & 1 \end{bmatrix}$$

- 10-8 Range: Horizontal and slant.** A radar uses the measured slant range  $R_m$  of an object and uses a “nominal” (hypothesized) altitude  $h_0$  to convert it into horizontal range

$$r = \sqrt{R_m^2 - h_0^2}$$

1. Assuming that the measured slant range has an error  $\tilde{R} \sim (0, \sigma_R^2)$  and that the error in the assumed altitude is  $\tilde{h} \sim (0, \sigma_h^2)$ , find the first two moments of the error  $\tilde{r}$  in the horizontal range. Indicate explicitly the assumptions used.
2. Evaluate the standard deviation  $\sigma_r$  for  $R = 10^5 \text{ m}$ ,  $\sigma_R = 10^2 \text{ m}$ ,  $h = 2 \cdot 10^3 \text{ m}$ ,  $\sigma_h = 10^3 \text{ m}$ .
3. State your opinion about the significance of the effect of the altitude error at this range. What happens at a shorter range  $R = 10^4 \text{ m}$ ?

- 10-9 RMS speed and velocity errors in Monte Carlo simulations.** Indicate the procedure to evaluate the RMS speed and velocity errors in Monte Carlo simulations using the velocity component estimates and their true values.

- 10-10 Geometric Dilution of Precision (GDOP) in Global Positioning System (GPS).** Geometric Dilution of Precision is an important problem in the Global Positioning System, which obtains the position of an observer based on (pseudo)range measurements from a constellation of satellites. It is defined as the degradation of the overall position estimate RMSE vs. the ranging accuracy (RMSE) from the satellites.

Consider the following simplified scalar version of this problem.

The position  $x$  of a point on the  $x$ -axis is sought based on one range measurement from a known location  $(x_1, y_1)$  with accuracy (RMSE)  $\sigma_r$ .

1. Assuming small errors ( $\sigma_r \ll r$ ) and a nominal location of the observer  $x = x_0$ , find the GDOP  $\sigma_x/\sigma_r$ .
2. Calculate the GDOP for  $y_1 = 10^5 \text{ km}$  and  $|x_0 - x_1| = 10^4 \text{ km}$ .
3. Calculate the GDOP for  $y_1 = 10^5 \text{ km}$  and  $|x_0 - x_1| = 10^6 \text{ km}$ .

Next consider the following simplified vector version of this problem.

The position  $\mathbf{p}$  of a point in space is sought based on several range measurements  $\mathbf{r} = [r_1 \dots r_n]'$  from known locations  $\mathbf{X} = \{\mathbf{x}_i, i = 1, \dots, n\}$ . The measurement is

$$\mathbf{r} = \psi(\mathbf{p}, \mathbf{X}) + \tilde{\mathbf{r}}$$

where  $\tilde{\mathbf{r}} \sim \mathcal{N}(0, \sigma_r^2 I)$  is the measurement error.

4. Assuming a nominal location of the observer  $\mathbf{p} = \mathbf{p}_0$ , find the GDOP defined as  $\text{RMS}(\hat{\mathbf{p}})/\sigma_r$ , where  $\hat{\mathbf{p}}$  is the estimation error of  $\mathbf{p}$ .

5. Calculate the GDOP for a planar case with  $\mathbf{p}_0 = [0 \ 0]'$ ,  $\mathbf{x}_1 = [-10^4 \ 10^5]'$ ,  $\mathbf{x}_2 = [10^4 \ 10^5]'$ .

- 10-11 Longitudinal and lateral acceleration.** Given the estimate of the state of an object moving in a plane  $\mathbf{x} = [x \ \dot{x} \ \ddot{x} \ y \ \dot{y} \ \ddot{y}]'$  (denoted without hats for simplicity) and the associated  $(6 \times 6)$  covariance matrix  $P$ .

- Find the magnitude of the (estimated) longitudinal acceleration vector  $a_{\text{long}}$ , i.e., the component of the (estimated) acceleration  $a = [\ddot{x} \ \ddot{y}]'$  along the (estimated) velocity vector  $v = [\dot{x} \ \dot{y}]'$ .
- Find the magnitude of the (estimated) lateral acceleration vector  $a_{\text{lat}}$ , i.e., the component of the (estimated) acceleration orthogonal to the (estimated) velocity vector.
- Find the magnitude of estimate of the turning rate  $\omega$  (ratio of the lateral acceleration magnitude to the speed [that is, to the velocity magnitude]).
- Using a first-order expansion find the variance of the estimated turn rate (derive the Jacobian  $\partial\omega/\partial\mathbf{x}$  and express the variance in terms of this and  $P$ ).

- 10-12 Radar cross-section estimation.** We want to estimate the radar cross-section (RCS)  $\rho$  of a target by illuminating it with an electronically scanned array radar and measuring the intensity of the return. However, the center (boresight) of the beam pointed toward the target is not pointed perfectly, resulting in less energy returned due to the antenna pattern. Thus a straightforward averaging of the RCS measurements based on the received reflection intensities will underestimate the target RCS.

Assuming a  $\cos^2$  antenna pattern (which becomes  $\cos^4$  for two-way propagation) and an azimuth pointing error  $\alpha$  (the off-boresight pointing angle — where the target is from the center of the beam), the measured RCS is

$$\rho_m = \rho \cos^4 \left( \frac{\alpha}{0.5\alpha_{BW}} \alpha_0 \right)$$

where  $\alpha_{BW}$  is the (two-sided) beamwidth in azimuth defined such that if the pointing error is  $0.5\alpha_{BW}$  — that is, half the beamwidth away from the center — then one has half the power in the return ( $\cos^4 \alpha_0 = 0.5$ , i.e.,  $\alpha_0 = 0.57 \text{ rad}$ ).

The azimuth pointing error is a r.v. with  $E\alpha = 0$ ,  $\sigma_\alpha = \alpha_{BW}/6$ . Similarly (and independently), the error in elevation  $\epsilon$  is a r.v. with  $E\epsilon = 0$ ,  $\sigma_\epsilon = \epsilon_{BW}/6$ .

Find  $E[\rho_m/\rho]$  accounting for both the azimuth and elevation losses due to mispointing according to the above models.

**10-13 Distance estimation in sailing.** An old mariner's technique to estimate the distance from a sailing vessel to a fixed point uses the equation

$$d = \frac{s\tau}{\alpha_2 - \alpha_1}$$

where  $s$  is the speed of the vessel, and  $\tau$  is the elapsed time over which the bearing angle of the point (as seen from the vessel) changes from  $\alpha_1$  to  $\alpha_2$  (from two compass readings).

Assume the following:

- The motion of the vessel is such that the point is on the perpendicular bisector of the straight line traveled by the vessel at constant speed.
- The indicated speed has a zero-mean additive error with s.d.  $\sigma_s$ .
- Each compass reading has a zero-mean additive error with s.d.  $\sigma_\alpha$ .
- The elapsed time has a zero-mean additive error with s.d.  $\sigma_\tau$ .

With this,

- Find the s.d.  $\sigma_d$  of the error in the estimate  $d$ . Indicate what additional assumptions are necessary.
- Evaluate  $\sigma_d$  and the ratio  $\sigma_d/d$  for  $s = 7$  nmi/hr,  $\sigma_s = 0$ ,  $\tau = 3$  min,  $\sigma_\tau = 0$ ,  $\alpha_1 = 270^\circ$ ,  $\alpha_2 = 280^\circ$ ,  $\sigma_\alpha = 2^\circ$ .

### 10.8.3 Project — Nonlinear Filtering with Angle-Only Measurements

Solve the problem of tracking with an angle-only sensor from Subsection 10.3.4 with the numerical technique of nonlinear filtering discussed in Subsection 10.2.5.

Define a grid in the state space that varies with the (estimated) motion of the target.

Define the procedure for choosing the support of the pdf, i.e., the extreme values of the set of points over which the pdf is evaluated (the intervals in position and velocity).

First, try a grid with uniformly spaced points. Then, using an appropriate piecewise analytical function, design a grid with points spaced at higher density where the pdf has more mass, and compare the performance with that obtained using the uniformly spaced grid.

Note that there is compromise between choosing long intervals with many points (which require many computations) and shorter intervals with fewer points (which degrade the performance). Indicate the integration technique used and make sure the pdfs obtained integrate to unity.

- Plot, for a single run, the evolution in time of
  - the pdf of the position, the estimate (conditional mean), the  $\pm 3\sigma$  points and the true value
  - the pdf of the velocity, the estimate (conditional mean), the  $\pm 3\sigma$  points and the true value
- Plot for 50 Monte Carlo runs
  - the NEES
  - the NIS

- (iii) the RMS position error
- (iv) the RMS velocity error

for the nonlinear filter and for the linearized filter (with the same random numbers).

Compare the results for four different grids indicating the number of grid points and their location (uniform/nonuniform).

# Chapter 11

## ADAPTIVE ESTIMATION AND MANEUVERING TARGETS

### 11.1 INTRODUCTION

#### 11.1.1 Adaptive Estimation — Outline

In the models considered previously, the only uncertainties consisted of additive white noises (process and measurement) with known statistical properties. In other words, the system model, consisting of the state transition matrix, the input gain (and input, if any), the measurement matrix, and noise covariances, were all assumed to be known. In the case of “colored” noise with known autocorrelation, it has been shown that it can be reduced to the standard situation in Chapter 8.

In many practical situations the above listed “parameters of the problem” are partially unknown and possibly time-varying. There are numerous techniques of *system identification* (e.g., [Ljung87]) that deal with the identification (estimation) of the *structural parameters* of a system.

The purpose of this chapter is to present state estimation techniques that can “adapt” themselves to certain types of uncertainties beyond those treated earlier — *adaptive estimation algorithms*.

One type of uncertainty to be considered is the case of unknown inputs into the system, which typifies maneuvering targets. The other type will be a combination of system parameter uncertainties with unknown inputs where the system parameters (are assumed to) take values in a discrete set.

Maneuvering targets are characterized by an equation of the same form as (4.3.1-9), namely,

$$x(k+1) = F(k)x(k) + G(k)u(k) + v(k) \quad (11.1.1-1)$$

but the input  $u(k)$ , which enters the system in addition to the process noise  $v(k)$ , is *unknown*.

For simplicity, linear models are considered — in the case of nonlinear models the same techniques that are discussed in the sequel can be used with linearization.

The approaches that can be used in such a situation fall into two broad categories:

1. The unknown input (the maneuver command) is modeled as a random process.
2. The unknown input is estimated in real time.

The random process type models can be classified according to the statistical properties of the process modeling the maneuver as

- White noise or
- Autocorrelated noise

Note that this *unknown input as a random process* approach amounts to treating the unknown input as (an additional) process noise, and, possibly, requiring augmentation of the system's state.

The use of fixed-level white noise to model maneuvers has been discussed in the context of kinematic models. However, maneuvers, by their nature, are of different magnitudes at different times. One way to accommodate this is by adjusting the level of the process noise. One option is a *continuous noise level adjustment* technique, discussed in Section 11.2. Alternatively, several discrete levels of noise can be assumed in the filter, with a *noise level switching* according to a certain rule, also discussed in Section 11.2.

The use of autocorrelated (colored) noise to model maneuvers has been presented in Section 8.2 — the approach is to “prewhiten” the noise and augment the state with the prewhitening subsystem. The noise level adjustment techniques can be used with the augmented system, which is driven by white noise.

All these methods relying on *modeling maneuvers as random processes* are approximations because maneuvers are, in general, not stochastic processes. Nevertheless, such approaches are simple and can be quite effective.

The second type of approach, *input estimation*, is implemented assuming the input to be constant over a certain period of time. The estimation can be done based on the least squares criterion, and the result can be used in the following ways:

- The state estimate is corrected with the estimated input, or
- The state is augmented and the input becomes an extra state component that is reestimated sequentially within the augmented state.

The input estimation with state estimate correction technique is presented in Section 11.3. The technique of estimating the input and, when “statistically significant,” augmenting the state with it (which leads to *variable state dimension*), is the topic of Section 11.4. These two algorithms and the noise level switching technique of Section 11.2 are compared in Section 11.5.

The so-called *multiple model (MM)* algorithms are the topic of Section 11.6. These algorithms assume that the system behaves according to one of a finite number of models — it is in one of several *modes* (operating regimes). The models can differ in *noise levels* or their *structure* — different state dimensions and unknown inputs can be accommodated as well. Such systems are called *hybrid systems* — they have both discrete (structure/parameters) and continuous uncertainties (additive noises).

First the *static MM* algorithm — static from the point of view of the assumed evolution of the models, i.e., for fixed (nonswitching) models — is considered. Then the optimal *dynamic* algorithm — for switching models, according to a Markov chain — is presented. Since the optimal dynamic algorithm is not practical for implementation, two suboptimal approaches, one called *generalized pseudo-Bayesian (GPB)* and the other the *interacting multiple model (IMM)*, are also presented.

The design of an IMM estimator for *air traffic control (ATC)* is discussed in detail. Guidelines are also developed for when an adaptive estimator is really needed, i.e., when a (single model based) Kalman filter is not adequate.

The use of the extended Kalman filter for state and system parameter estimation is briefly presented in Section 11.9.

### 11.1.2 Adaptive Estimation — Summary of Objectives

Show modeling of maneuvers as

- A random process (white or autocorrelated process noise)
  - continuously variable level;
  - several discrete levels with switching between them.
- A fixed input, estimated as an unknown constant, with
  - state estimate compensation;
  - variable state dimension.

Compare some of these algorithms.

Illustrate the optimal method of comparison of algorithms.

Multiple-model (hybrid system) approach:

Several models are used in parallel with a probabilistic weighting:

- Static — for fixed (nonswitching) models
- Dynamic — for switching models
  - Optimal
  - GPB1
  - GPB2
  - IMM

Present the design of an IMM estimator for air traffic control.

Answer the question: When is an adaptive estimator really needed?

Show the use of the EKF for state and system parameter estimation.

## 11.2 ADJUSTABLE LEVEL PROCESS NOISE

### 11.2.1 Continuous Noise Level Adjustment

In this *continuous noise level adjustment* approach, the target is tracked with a filter in which a certain low level of process noise is assumed. The level of the noise is determined by its variance (or covariance matrix).

A maneuver manifests itself as a “large” innovation. A simple detection procedure for such an occurrence is based on the *normalized innovation squared*

$$\epsilon_\nu(k) = \nu(k)' S(k)^{-1} \nu(k) \quad (11.2.1-1)$$

A threshold is set up such that based on the target model (for the nonmaneuvering situation)

$$P\{\epsilon_\nu(k) < \epsilon_{max}\} = 1 - \mu \quad (11.2.1-2)$$

with, say, tail probability  $\mu = 0.01$ .

If the threshold is exceeded, then the process noise variance  $Q(k-1)$  is scaled up until  $\epsilon_\nu$  is reduced to the threshold  $\epsilon_{max}$ .

Using a *scaling factor*  $\phi(k)$  (i.e., a *fudge factor*), the covariance of the innovations becomes

$$S(k) = H(k)[F(k-1)P(k-1|k-1)F(k-1)' + \phi(k)Q(k-1)]H(k)' + R(k) \quad (11.2.1-3)$$

In place of the single-time test statistic (11.2.1-1) one can also use a time average over a “sliding window.” This is illustrated in the next subsection.

A similar technique can be used to lower the process noise level after the maneuver.

### 11.2.2 Process Noise with Several Discrete Levels

Another approach is to assume two or more levels of process noise and use a rule for *noise level switching*.

Under normal conditions the filter is operating with the low level noise  $Q_1$ .

The normalized innovation squared

$$\epsilon_\nu(k) = \nu(k)' S(k)^{-1} \nu(k) \quad (11.2.2-1)$$

is monitored; and if it *exceeds* a certain threshold, the filter switches to a prechosen higher level of process noise,  $Q_2$ .

Under the linear-Gaussian assumptions, the pdf of  $\epsilon_\nu$  is chi-square distributed with  $n_z$ , the dimension of the measurement, degrees of freedom:

$$\epsilon_\nu \sim \chi_{n_z}^2 \quad (11.2.2-2)$$

The switching threshold is chosen based on (11.2.2-2) such that the probability of being exceeded under normal conditions is small.

The decision statistic (11.2.2-1), which is based on a single sampling time, can be replaced by a ***moving average*** (or ***moving sum***) of the normalized innovations squared over a ***sliding window*** of  $s$  sampling times

$$\epsilon_\nu^s(k) = \sum_{j=k-s+1}^k \epsilon_\nu(j)$$

(11.2.2-3)

The above is chi-square distributed with  $sn_z$  degrees of freedom

$$\epsilon_\nu^s \sim \chi_{sn_z}^2 \quad (11.2.2-4)$$

since (11.2.2-3) is the sum of  $s$  independent terms with distribution (11.2.2-2).

Alternatively, a ***fading memory average*** (also called ***exponentially discounted average***)

$$\epsilon_\nu^\alpha(k) = \alpha \epsilon_\nu^\alpha(k-1) + \epsilon_\nu(k)$$

(11.2.2-5)

where

$$0 < \alpha < 1 \quad (11.2.2-6)$$

and with initial condition  $\epsilon_\nu^\alpha(0) = 0$ , can be used.

The expected value of (11.2.2-5) in steady state is

$$E[\epsilon_\nu^\alpha(k)] = \frac{n_z}{1-\alpha} \quad (11.2.2-7)$$

The ***effective window length*** of (11.2.2-5) can be considered as the sum of the weights multiplying  $\epsilon_\nu^\alpha(k)$ ,  $k = 1, 2, \dots$ , i.e.,

$$s_\alpha = 1 + \alpha + \alpha^2 + \dots = \frac{1}{1-\alpha} \quad (11.2.2-8)$$

For example, for  $\alpha = 0.8$  one has  $s_\alpha = 5$ .

Based on (11.2.2-7), one can assume, as a first approximation (matching the first moment only), that  $\epsilon_\nu^\alpha(k)$  is chi-square distributed with number of degrees of freedom given by (11.2.2-7).

Using the first and second moment-matching approximation described in Subsection 1.4.18, one obtains

$$\epsilon_\nu^\alpha \sim \frac{1}{1+\alpha} \chi_{n_\alpha}^2 \quad (11.2.2-9)$$

where the number of degrees of freedom is

$$n_\alpha = n_z \frac{1 + \alpha}{1 - \alpha} \quad (11.2.2-10)$$

For example, for  $\alpha = 0.8$  one obtains  $n_\alpha = 18$ . Using the first approximation for the same  $\alpha$  with  $n_z = 2$  leads to 10 degrees of freedom. In other words, the moment-matching approximation (with the first two moments) indicates a (relatively) narrower pdf. The “width” of a chi-square pdf can be taken as the ratio of its standard deviation to its mean. For  $\chi_n^2$ , this is (see Subsection 1.4.17)  $\sqrt{2n}/n$ , which decreases as  $n$  increases.

### Estimator Operation

Thus, the filter switches from the lower process noise covariance  $Q_1$  to the higher  $Q_2$  if the average (11.2.2-3) or (11.2.2-5) exceeds an *upcrossing threshold*, determined based on (11.2.2-4) or (11.2.2-9), respectively, with a small tail probability.

After the filter starts running with the higher-level process noise the innovations are monitored again to see whether there is reason to switch back.

The change to the model with lower-level process noise is done when the normalized innovation, or a certain average of it, falls *below* another threshold — the *downtcrossing threshold*.

There is no exact way to choose these thresholds — even if the distributions were known exactly, the tail probabilities are still subjective. They can be chosen, to begin with, based on tail probabilities and then set following experimentation (Monte Carlo runs) and subjective evaluation of the results.

### Remarks

This technique of noise level switching can be extended to more than two levels of process noise. It can also be used with white process noise or autocorrelated process noise; the latter has to be prewhitened, as discussed in Subsection 8.2.1.

In general,  $n_\alpha$ , given in (11.2.2-10), is not an integer. In this case one has a gamma rather than chi-square distribution; for the sake of simplicity, however, one can use the chi-square tables with an interpolation.

### 11.2.3 Adjustable Level Process Noise — Summary

A maneuver can be detected by monitoring the *normalized innovation*; if it exceeds a threshold, then it can be assumed (in the absence of other factors) that the target has deviated from its previous “pattern.”

A *fudge factor* can be used to scale up (continuously) the process noise (at the previous time) such that the modified prediction covariance is sufficiently

large for the normalized innovation to be below the set threshold; the increased innovation covariance should “cover” the observed deviation.

Alternatively, a number of levels of process noise can be assumed. In this approach, at any given time a *single filter* operates with the “*current*” noise level.

A simple rule of switching from the current noise level to the one above or below is followed.

The innovations are monitored via

- a moving average over a sliding window or
- a fading-memory average

of their normalized squared value.

This average is compared to a threshold based on the chi-square density for a small upper tail probability. If this *upcrossing threshold* is exceeded, then the filter switches to a higher level of process noise.

The downswitch is done using a *downcrossing threshold*.

## 11.3 INPUT ESTIMATION

### 11.3.1 The Model

Consider the system with state equation

$$x(k+1) = Fx(k) + Gu(k) + v(k) \quad (11.3.1-1)$$

where  $u$  is an *unknown input* modeling the target maneuvers and  $v$  is the process noise, zero mean white with known covariance  $Q$ .

The observations are

$$z(k) = Hx(k) + w(k) \quad (11.3.1-2)$$

with the observation noise  $w$  zero mean, white, with covariance  $R$ , and independent of the process noise.

The estimation of the state is done using the model without input (nonmaneuvering)

$$x(k+1) = Fx(k) + v(k) \quad (11.3.1-3)$$

Two Kalman filters are considered:

1. The *actual one* based on the nonmaneuvering model (11.3.1-3).
2. A *hypothetical one* based on the maneuvering model (11.3.1-1) with known  $u$ .

From the innovations of the *nonmaneuvering filter* based on (11.3.1-3), the input  $u$  is to be

- detected,
- estimated, and
- used to correct the state estimate.

This will be done using a sliding window of the latest  $s$  measurements, and during this window period the input will be assumed *constant*.

### 11.3.2 The Innovations as a Linear Measurement of the Unknown Input

Denote the present time by  $k$  and assume that the target starts maneuvering at time  $k - s$ ; that is, the **maneuver onset time** is  $k - s$ . The unknown inputs during the interval  $[k - s, \dots, k]$  are  $u(i)$ ,  $i = k - s, \dots, k - 1$ .

The state estimates from the *mismatched nonmaneuvering filter* based on (11.3.1-3) will be denoted by an asterisk. The recursion for these estimates is

$$\begin{aligned}\hat{x}^*(i+1|i) &= F[I - W(i)H]\hat{x}^*(i|i-1) + FW(i)z(i) \\ &\triangleq \Phi(i)\hat{x}^*(i|i-1) + FW(i)z(i) \\ &\quad i = k - s, \dots, k - 1\end{aligned}\quad (11.3.2-1)$$

with the initial condition

$$\hat{x}^*(k-s|k-s-1) = \hat{x}(k-s|k-s-1) \quad (11.3.2-2)$$

being the correct estimate (one-step prediction) before the maneuver started. This follows from the assumption that the maneuver onset time is  $k - s$ .

Similarly to (4.3.3-1), recursion (11.3.2-1) yields, in terms of (11.3.2-2),

$$\begin{aligned}\hat{x}^*(i+1|i) &= \left[ \prod_{j=0}^{i-k+s} \Phi(i-j) \right] \hat{x}(k-s|k-s-1) \\ &\quad + \sum_{j=k-s}^i \left[ \prod_{m=0}^{i-j-1} \Phi(i-m) \right] FW(j)z(j) \quad i = k - s, \dots, k - 1\end{aligned}\quad (11.3.2-3)$$

If the inputs were known, the *hypothetical correct filter* based on (11.3.1-1) would yield estimates according to the recursion

$$\begin{aligned}\hat{x}(i+1|i) &= \Phi(i)\hat{x}(i|i-1) + FW(i)z(i) + Gu(i) \\ &= \left[ \prod_{j=0}^{i-k+s} \Phi(i-j) \right] \hat{x}(k-s|k-s-1) \\ &\quad + \sum_{j=k-s}^i \left[ \prod_{m=0}^{i-j-1} \Phi(i-m) \right] [FW(j)z(j) + Gu(j)] \\ &\quad i = k - s, \dots, k - 1\end{aligned}\quad (11.3.2-4)$$

which is the same as (11.3.2-3) except for the last term containing the inputs.

The innovations

$$\nu(i+1) = z(i+1) - H\hat{x}(i+1|i) \quad (11.3.2-5)$$

corresponding to the correct (but *hypothetical*) filter (11.3.2-4) are a zero-mean white sequence with covariance  $S(i+1)$  given as in (5.2.3-9).

The innovations corresponding to the *nonmaneuvering filter* (11.3.2-3) are

$$\nu^*(i+1) = z(i+1) - H\hat{x}^*(i+1|i) \quad (11.3.2-6)$$

From (11.3.2-3) and (11.3.2-4) it follows that the innovations (11.3.2-6) of the nonmaneuvering filter — the *only real filter* — are the same as the “white noise sequence” (11.3.2-5) plus a “bias term” related to the inputs

$$\nu^*(i+1) = \nu(i+1) + H \sum_{j=k-s}^i \left[ \prod_{m=0}^{i-j-1} \Phi(i-m) \right] Gu(j) \quad (11.3.2-7)$$

Assuming the input to be constant over the interval  $[k-s, \dots, k-1]$ , that is,

$$u(j) = u \quad j = k-s, \dots, k-1 \quad (11.3.2-8)$$

yields

$$\boxed{\nu^*(i+1) = \Psi(i+1)u + \nu(i+1) \quad i = k-s, \dots, k-1} \quad (11.3.2-9)$$

where

$$\Psi(i+1) \triangleq H \sum_{j=k-s}^i \left[ \prod_{m=0}^{i-j-1} \Phi(i-m) \right] G \quad (11.3.2-10)$$

Equation (11.3.2-9) shows that

the *nonmaneuvering filter innovation*  $\nu^*$

is

a *linear measurement of the input*  $u$

in the presence of the additive “white noise”  $\nu$  given by (11.3.2-5) above.

### 11.3.3 Estimation of the Unknown Input

Based on (11.3.2-9), the input can be estimated via LS from

$$y = \Psi u + \epsilon \quad (11.3.3-1)$$

where

$$y = \begin{bmatrix} \nu^*(k) \\ \vdots \\ \nu^*(k-s+1) \end{bmatrix} \quad (11.3.3-2)$$

is the stacked “measurement” vector,

$$\Psi = \begin{bmatrix} \Psi(k) \\ \vdots \\ \Psi(k-s+1) \end{bmatrix} \quad (11.3.3-3)$$

is the measurement matrix, and the “noise”  $\epsilon$ , whose components are the innovations (11.3.2-5), is zero mean with block-diagonal covariance matrix

$$S = \text{diag}[S(i)] \quad (11.3.3-4)$$

The **input estimate** in batch form is

$$\hat{u} = (\Psi' S^{-1} \Psi)^{-1} \Psi' S^{-1} y \quad (11.3.3-5)$$

with the resulting covariance matrix

$$L = (\Psi' S^{-1} \Psi)^{-1} \quad (11.3.3-6)$$

An estimate is accepted, that is, a **maneuver detection** is declared if and only if (11.3.3-5) is *statistically significant* (see Subsection 1.5.2). The significance test for the estimate  $\hat{u}$ , which is an  $n_u$ -dimensional vector, is

$$\epsilon_{\hat{u}} = \hat{u}' L^{-1} \hat{u} \geq c \quad (11.3.3-7)$$

where  $c$  is a threshold chosen as follows. If the input is zero, then

$$\hat{u} \sim \mathcal{N}(0, L) \quad (11.3.3-8)$$

The statistic  $\epsilon_{\hat{u}}$  from (11.3.3-7) is then chi-square distributed with  $n_u$  degrees of freedom, and  $c$  is such that the probability of false alarm is

$$P\{\epsilon_{\hat{u}} \geq c\} = \alpha \quad (11.3.3-9)$$

with, e.g.,  $\alpha \leq 0.01$ .

### 11.3.4 Correction of the State Estimate

If a maneuver is detected, then the state estimate has to be corrected. This is done, as before, assuming that the maneuver onset time was  $k-s$ . The term reflecting the effect of the input in (11.3.2-4) is used to correct the predicted state as follows

$$\hat{x}^u(k|k-1) = \hat{x}^*(k|k-1) + M(k)\hat{u} \quad (11.3.4-1)$$

where

$$M(k) \triangleq \sum_{j=k-s}^{k-1} \left[ \prod_{m=0}^{k-j-2} \Phi(k-1-m) \right] G \quad (11.3.4-2)$$

The covariance associated with the prediction (11.3.4-1) is

$$P^u(k|k-1) = P(k|k-1) + M(k)L M(k)' \quad (11.3.4-3)$$

where  $L$  is given in (11.3.3-6). Note that the covariance increases — the last term above is positive semidefinite or positive definite — to account for the uncertainty in the correction term used in (11.3.4-1).

The assumption about the maneuver onset time being  $k-s$  at every  $k$  makes the algorithm very elegant, but not optimal.

## Maneuver Onset Time Estimation

It is possible to estimate the onset time using a maximum likelihood criterion — this requires running in parallel a number of such algorithms. Each algorithm evaluates the likelihood function for its assumed maneuver onset time and the most likely effective window length  $s$  is chosen.

It is clear that this becomes quite expensive.

### 11.3.5 Input Estimation — Summary

The input estimation method is based on the following:

Given a linear system driven by a constant input, a filter based on the state model of the system *without input (input equal to zero)* estimates the state.

Then the innovations of this *mismatched filter* are

- a linear measurement of the input
- with additive zero-mean white noise with covariance equal to the filter's innovation covariance.

This allows the estimation of the input, assuming that

- it started at the beginning of the assumed sliding window, and
- it is approximately constant over this window in time.

The estimation is done via least squares.

The estimated input (e.g., acceleration) is tested for *statistical significance* (i.e., if it is “large enough” compared to its standard deviation).

If the estimated input is statistically significant, then the state estimate is corrected with the effect of the input from the time it was assumed to have started (at the beginning of the window).

The covariance of the state is adjusted (increased) because the correction was done with an estimated input, rather than the exact input, which is not available.

## 11.4 THE VARIABLE STATE DIMENSION APPROACH

### 11.4.1 The Approach

The target maneuver is considered here an inherent part of its dynamics, rather than noise. In the absence of maneuvers the filter operates using a “quiescent state model.” Once a maneuver is detected, new state components are added — thus the name *variable state dimension (VSD)* approach.

The extent of the maneuver as detected is then used to yield an estimate for the extra state components, and corrections are made on the other state

components. The tracking is then done with the augmented state model until it will be reverted to the quiescent model by another decision.

The rationale for using a lower-order quiescent model and a higher-order maneuvering model is the following: This will allow good tracking performance in both situations rather than a compromise. For example, if the target does not have acceleration, using a third-order model increases the estimation errors for both position and velocity.

The two models used here, described in Section 6.3, are a (nearly) constant velocity model for the quiescent situation and a (nearly) constant acceleration model for the maneuvering situation.

The state vector, for a planar motion, is in the *quiescent model*

$$x = [\xi \ \dot{\xi} \ \eta \ \dot{\eta}]' \quad (11.4.1-1)$$

In the *maneuvering model*, the state is

$$x^m = [\xi \ \dot{\xi} \ \eta \ \dot{\eta} \ \ddot{\xi} \ \ddot{\eta}]' \quad (11.4.1-2)$$

The method is not limited to these models. For example, the maneuvering situation could be modeled using autocorrelated acceleration, as in Section 8.2.

### 11.4.2 The Maneuver Detection and Model Switching

The target tracking filter is initialized under the constant velocity model assumption as in Section 5.5, using a two-point differencing procedure.

The maneuver detection is done as follows:

A *fading memory average* of the innovations from the estimator based on the quiescent model is computed

$$\epsilon_v^\alpha(k) = \alpha \epsilon_v^\alpha(k-1) + \epsilon_v(k) \quad (11.4.2-1)$$

with

$$\epsilon_v(k) \triangleq \nu(k)' S(k)^{-1} \nu(k) \quad (11.4.2-2)$$

where  $0 < \alpha < 1$ ,  $\nu(k)$  is the innovation and  $S(k)$  its covariance.

Since  $\epsilon_v(k)$  is, under the Gaussian assumptions, chi-square distributed with  $n_z$  (dimension of the measurement) degrees of freedom, one has

$$\lim_{k \rightarrow \infty} E[\epsilon_v^\alpha(k)] = \frac{n_z}{1 - \alpha} \quad (11.4.2-3)$$

As in Section 11.2, one can look at

$$s_\alpha = \frac{1}{1 - \alpha} \quad (11.4.2-4)$$

as the *effective window length* over which the presence of a maneuver is tested.

The hypothesis that a maneuver is taking place is accepted if  $\epsilon_{\nu}^{\alpha}(k)$  exceeds a certain threshold that can be determined based on the chi-square distribution as in Section 11.2. Then the estimator switches from the quiescent model to the maneuvering model.

The scheme for reverting to the quiescent model is as follows: The estimated accelerations are compared to their variances, and if they are not statistically significant the maneuver hypothesis is rejected.

### Maneuver Termination vs. Maneuver Onset Detection

It is more difficult to detect *maneuver termination* than *maneuver onset* because a maneuvering model has a larger innovation covariance than a nonmaneuvering model. This is due to the fact that the latter has a larger state vector and assumes more motion uncertainty than the former.

### Significance Test for the Acceleration Estimate

The test statistic for significance of the acceleration estimates is

$$\epsilon_{\hat{a}}(k) = \hat{a}(k|k)' P_a^m(k|k)^{-1} \hat{a}(k|k) \quad (11.4.2-5)$$

where  $\hat{a}$  is the estimate of the acceleration components and  $P_a^m$  is the corresponding block from the covariance matrix of the maneuvering model.

When the *moving average* over a window of length  $p$

$$\epsilon_{\hat{a}}^p(k) = \sum_{j=k-p+1}^k \epsilon_{\hat{a}}(j) \quad (11.4.2-6)$$

falls below a threshold, the acceleration is deemed *insignificant*. A fading memory average can also be used.

The situation where the acceleration drops suddenly to zero can lead to large innovations for the maneuvering model. This can be taken care of by allowing a switch to the lower-order model also when the maneuvering model's innovation exceeds, say, the 99% confidence region.

### 11.4.3 Initialization of the Augmented Model

When a maneuver is detected at time  $k$ , the filter assumes that the target had a constant acceleration starting at  $k - s - 1$ , where  $s$  is the effective window length. The state estimates for time  $k - s$  are then modified as follows. It is assumed that only position measurements are available.

First, the estimates at  $k - s$  for the acceleration are

$$\hat{x}_{4+i}^m(k-s|k-s) = \frac{2}{T_2} [z_i(k-s) - \hat{z}_i(k-s|k-s-1)] \quad i = 1, 2 \quad (11.4.3-1)$$

The position components of the estimate at  $k - s$  are taken as the corresponding measurements, that is,

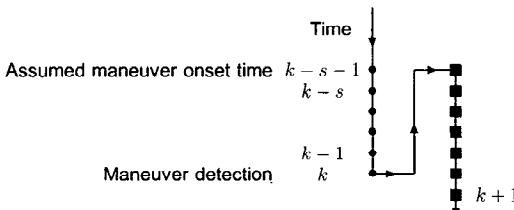
$$\hat{x}_{2i-1}^m(k-s|k-s) = z_i(k-s) \quad i = 1, 2 \quad (11.4.3-2)$$

while the velocity components are corrected with the acceleration estimates as follows:

$$\hat{x}_{2i}^m(k-s|k-s) = \hat{x}_{2i}(k-s-1|k-s-1) + T\hat{x}_{4+i}^m(k-s|k-s) \quad i = 1, 2 \quad (11.4.3-3)$$

The covariance matrix associated with the modified state estimate as given in (11.4.3-1) to (11.4.3-3) is  $P^m(k-s|k-s)$ , and its elements are presented in [Bar-Shalom82] together with their derivation. These equations specify in full the initialization of the filter based on the maneuvering model for time  $k-s$ .

Then, a recursive estimation algorithm (Kalman filter) based on this model reprocesses the measurements sequentially up to time  $k$ . Following this, the measurements are processed sequentially as they arrive. This is depicted in Fig. 11.4.3-1.



- Sequential estimation with quiescent model; ■ with maneuvering model

**Figure 11.4.3-1:** Switching from quiescent to maneuvering model.

#### 11.4.4 VSD Approach — Summary

In the absence of maneuvers, the filter is based on a *quiescent* (low-order) model of the state.

If a maneuver is detected, the estimator switches to a higher dimension (augmented) *maneuvering* model that incorporates additional state components (acceleration).

The use of the lower-order model yields maximum estimation accuracy in the case where the target undergoes no accelerations — no information is “wasted” on estimating state components that are zero.

A maneuver is declared detected when a *fading memory average* of the normalized innovations exceeds a threshold. The fading memory has an *effective*

*window length*, and the onset of the maneuver is then taken as the beginning of this sliding window.

The augmented filter is initialized at the beginning of the maneuver detection window.

A maneuver is declared terminated when the extra state component's (acceleration) estimates become statistically insignificant.

## 11.5 A COMPARISON OF ADAPTIVE ESTIMATION METHODS FOR MANEUVERING TARGETS

### 11.5.1 The Problem

In this section the following methods of maneuver detection are illustrated and compared:

1. White process noise with two levels (Section 11.2)
2. Input estimation (Section 11.3)
3. Variable dimension filtering (Section 11.4)

The example considers a target whose position is sampled every  $T = 10$  s. The target is moving in a plane with constant course and speed until  $k = 40$  when it starts a slow  $90^\circ$  turn that is completed in 20 sampling periods. A second fast turn of  $90^\circ$  starts at  $k = 61$  and is completed in 5 sampling times.

The initial condition of the target, with state

$$x = [\xi \ \dot{\xi} \ \eta \ \dot{\eta}]' \quad (11.5.1-1)$$

is, with position and velocity units m and m/s, respectively

$$x(0) = [2000 \ 0 \ 10000 \ -15]' \quad (11.5.1-2)$$

The slow turn is the result of acceleration inputs

$$u_\xi = u_\eta = 0.075 \text{ m/s}^2 \quad 400 \text{ s} \leq t \leq 600 \text{ s} \quad (11.5.1-3)$$

and the fast turn has accelerations

$$u^\xi = u^\eta = -0.3 \text{ m/s}^2 \quad 610 \text{ s} \leq t \leq 660 \text{ s} \quad (11.5.1-4)$$

Note that the changes in velocity are 0.75 and 3 m/s per sampling period for the slow and fast turn, respectively.

The measurements are position only according to the equation

$$z(k) = [1 \ 0 \ 1 \ 0] x(k) + w(k) \quad (11.5.1-5)$$

with  $w(k)$  zero mean, white, independent of the process noise, and with variance

$$E[w(k)^2] = rI \quad (11.5.1-6)$$

where  $r = 10^4 \text{ m}^2$ .

### 11.5.2 The White Noise Model with Two Levels

The adaptive algorithm based on white process noise with switching between two levels for maneuver modeling used the following state representation per coordinate (same for each coordinate and decoupled between them)

$$x(k+1) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} T/2 \\ 1 \end{bmatrix} v(k) \quad (11.5.2-1)$$

with  $v(k)$  zero mean white with variance

$$E[v(k)^2] = q \quad (11.5.2-2)$$

Note that here  $v(k)$  represents the **velocity increment over a sampling period**.

The two levels of process noise were  $q_0$  for the quiescent period and  $q_1$  after a maneuver was detected.

A maneuver was declared detected if the normalized innovation squared (11.2.2-1) exceeded a threshold  $\epsilon_m$ . Then  $q_0$  was replaced by  $q_1$  in the filter. A maneuver was considered terminated when the normalized innovation squared, obtained from the filter with the higher  $q_1$ , fell below the same threshold  $\epsilon_m$  — for simplicity, the upcrossing and downcrossing thresholds were taken the same.

The value for the quiescent process noise was  $q_0 = 0$ . Several values for the variance  $q_1$  of the process noise modeling the maneuver and the threshold  $\epsilon_m$  were tried. The best combination was determined based on the MSE in the position estimate of the target from 50 Monte Carlo runs. The resulting values were  $q_1 = 10$  and  $\epsilon_m = 3$ .

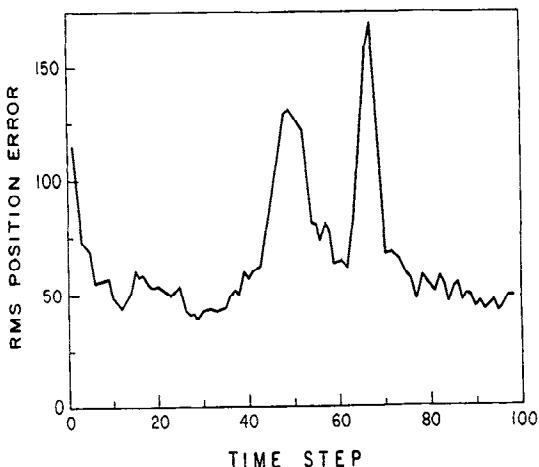
The RMS position error (coordinate  $\xi$ ) corresponding to these values is shown in Fig. 11.5.2-1. Note that the value  $\sqrt{q_1} \approx 3$  matches approximately the **maximum change in velocity per sampling period** during the fast maneuver, indicated in Subsection 11.5.1.

### 11.5.3 The IE and VSD Methods

The same problem was simulated using the **input estimation (IE)** and **variable state dimension (VSD)** methods.

The IE method was run with a window of  $s = 5$  samples to estimate the input. The state equation was the constant velocity model (6.3.2-1) for each coordinate and no process noise.

The VSD method was based on a quiescent model with state (11.4.1-1) and a maneuvering model with state (11.4.1-2). The quiescent model was, for each coordinate, the constant velocity state equation (6.3.2-1) with no process noise. The maneuvering model was a nearly constant acceleration state equation, given by (6.3.3-1) with process noise standard deviation  $\sigma_v$  in (6.3.3-4) taken equal to 5% of the estimated acceleration.



**Figure 11.5.2-1:** Position estimation error in coordinate  $\xi$  for the adaptive filter with two process noise levels (from 50 runs).

The following maneuver detection parameters were used in the VSD method:

1. The fading memory parameter from (11.4.2-1) was  $\alpha = 0.8$ , which corresponds to an effective window length (11.4.2-4) of  $s = 5$ .
2. The threshold for the fading memory average (11.4.2-1) was  $\chi_{10}^2(95\%) = 18.3$ . The maneuver detection was done together in the two coordinates based on both measurements (i.e.,  $n_z = 2$ ), which, with (11.4.2-3), yields that  $\epsilon_v^\alpha$  is approximately chi-square distributed with 10 degrees of freedom.
3. The window for the calculation of the average normalized acceleration (11.4.2-6) for the significance test was  $p = 2$ .
4. The threshold for the acceleration significance test was  $\chi_4^2(95\%) = 9.49$ , based on a 4-degree-of-freedom chi-square random variable (the dimension of the acceleration, 2, multiplied by  $p = 2$ ).

Figure 11.5.3-1a presents the position RMS errors (in coordinate  $\xi$ ) for the IE and VSD filters based on a 50-run Monte Carlo average. These two filters were run with the same random numbers for further comparison.

The RMS errors are smaller in the VSD filter compared to the IE filter. Comparing with Fig. 11.5.2-1, it is seen that the simple approach of two-level white process noise modeling of the maneuver is quite effective — it is slightly worse than the VSD filter but better than the IE filter.

Figure 11.5.3-1b presents the velocity RMS errors for the IE and VSD filters based on the same 50 Monte Carlo runs. For velocity, the VSD filter is only slightly superior compared to the IE algorithm.

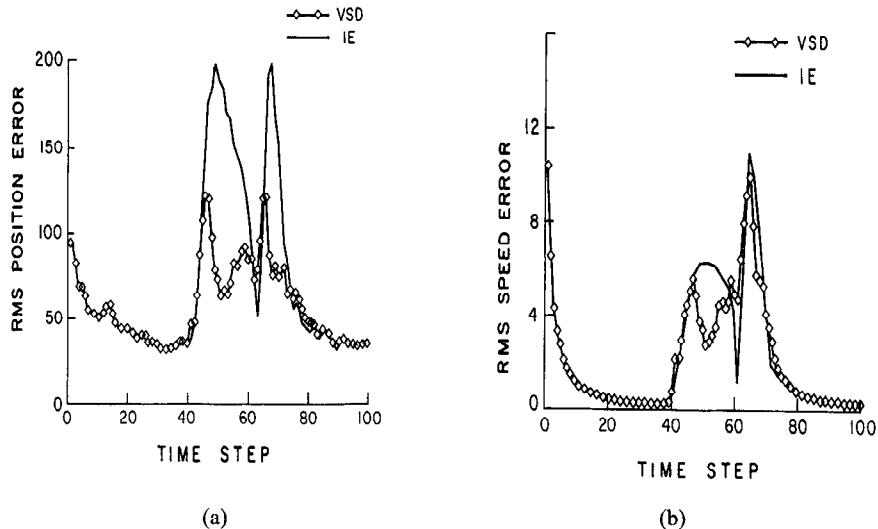


Figure 11.5.3-1: Estimation errors in coordinate  $\xi$  for the VSD and IE filters (from 50 runs).

### Remark

It should be noted that none of these three adaptive algorithms considered here manage to keep the RMS position error in the estimate below the position measurement noise RMS value (which is, in one coordinate, 100 m) *during the entire maneuver period*. In other words, these algorithms have difficulty providing **noise reduction** during the critical time of a maneuver.

This is a general problem with adaptive estimation algorithms: The adaptation might not be rapid enough. In the present example, the errors reach their peak shortly after the start of the maneuver.

This issue will be considered again in a later example.

### 11.5.4 Statistical Test for Comparison of the IE and VSD Methods

Next, we shall examine more closely the results of the IE and VSD filters and carry out the systematic comparison of algorithms presented in Subsection 11.5.3. The question is whether, in view of the limited sample size, one can state that the VSD algorithm is superior to the IE algorithm.

The performance of interest will be the mean square error in the estimate of one state component, not indicated explicitly, for simplicity

$$\epsilon^{\text{VSD}}(k) \triangleq E[(\hat{x}^{\text{VSD}}(k|k))^2] \quad (11.5.4-1)$$

$$\epsilon^{\text{IE}}(k) \triangleq E[(\hat{x}^{\text{IE}}(k|k))^2] \quad (11.5.4-2)$$

The **sample mean** of the performance from  $N$  Monte Carlo runs is

$$\bar{\epsilon}^{\text{VSD}}(k) = \frac{1}{N} \sum_{i=1}^N [\tilde{x}^{\text{VSD}i}(k|k)]^2 \quad (11.5.4-3)$$

$$\bar{\epsilon}^{\text{IE}}(k) = \frac{1}{N} \sum_{i=1}^N [\tilde{x}^{\text{IE}i}(k|k)]^2 \quad (11.5.4-4)$$

where  $\tilde{x}^{\text{VSD}i}(k|k)$  is the estimation error of the state component under consideration at time  $k$  in run  $i$  with algorithm VSD, and similarly for IE. These are the quantities shown in Figs. 11.5.3-1a and 11.5.3-1b.

We want to test if one can accept the hypothesis

$$H_1 : T^{\text{IV}}(k) \triangleq \bar{\epsilon}^{\text{IE}}(k) - \bar{\epsilon}^{\text{VSD}}(k) > 0 \quad (11.5.4-5)$$

that is, that algorithm VSD is superior to algorithm IE. Note that “superiority” is based on the *true mean square errors* defined in (11.5.4-1) and (11.5.4-2). A comparison of the *sample mean square errors* (11.5.4-3) and (11.5.4-4) corresponding to the two algorithms is not sufficient (actually is rather naive) since the inaccuracy in these sample means is not considered at all.

The correct statistical test is, as discussed in Subsection 1.5.3, based on the **sample performance differences**

$$T^{\text{IV}i}(k) \triangleq [\tilde{x}^{\text{IE}i}(k|k)]^2 - [\tilde{x}^{\text{VSD}i}(k|k)]^2 \quad (11.5.4-6)$$

The test uses the sample mean  $\bar{T}$  of the above differences and its standard error  $\sigma_{\bar{T}}$  given in (1.5.3-9) and (1.5.3-10), respectively.

If the ratio  $\bar{T}/\sigma_{\bar{T}}$  exceeds a threshold, then the *difference of the true means* in (11.5.4-5) is accepted as positive ( $H_1$  is accepted) and the null hypothesis

$$H_0 : T^{\text{IV}} \leq 0 \quad (11.5.4-7)$$

is rejected because it has a “low level of significance.” This is equivalent to saying that the estimated mean  $\bar{T}$  is positive and statistically significant (i.e.,  $H_1$  can be accepted because it is “significant”).

Hypothesis  $H_1$  is usually accepted only if the significance level of  $H_0$  is less than 5%, in which case the test is

$$\frac{\bar{T}}{\sigma_{\bar{T}}} > \mathcal{G}(95\%) = 1.65 \quad (11.5.4-8)$$

where  $\mathcal{G}(1 - \alpha)$  represents the point on the standard Gaussian distribution corresponding to upper tail probability of  $\alpha$ .

The above statistical test was applied for comparing the performance over intervals  $[k, l]$ , rather than single points in time, by replacing (11.5.4-6) with

$$T^{\text{IV}i}(k, l) = \frac{1}{l - k + 1} \sum_{m=k}^l \{[\tilde{x}^{\text{IE}i}(m|m)]^2 - [\tilde{x}^{\text{VSD}i}(m|m)]^2\} \quad (11.5.4-9)$$

Table 11.5.4-1 shows the test for the difference of the MSE between the two algorithms over several time intervals. Note that the slow maneuver took place during the time interval [40, 60] and that the fast one took place during the interval [61, 66]. Statistically significant improvements of the VSD algorithm over the IE algorithm are observed in position over all the intervals considered and in velocity over two intervals. In the remaining two intervals, while there was improvement, it was not sufficient to reject  $H_0$  at the 5% level. At the 8% level, however,  $H_0$  would have been rejected in all cases.

**Table 11.5.4-1:** Test of means for comparison of VSD and IE algorithms from 50 runs.

Time interval	Velocity			Position		
	$\bar{T}$	$\sigma_{\bar{T}}$	Test statistic	$\bar{T}$	$\sigma_{\bar{T}}$	Test statistic
40-60	8.06	1.19	6.77	13,560	735	18.45
60-70	4.99	3.28	1.52	10,336	1,013	10.20
40-70	7.72	1.43	5.40	12,677	619	20.48
60-80	2.45	1.75	1.40	6,091	626	9.73

## 11.5.5 Comparison of Several Algorithms — Summary

The optimal statistical method of comparing two algorithms based on Monte Carlo runs has been illustrated in the evaluation of the variable state dimension and input estimation filters.

The VSD filter turned out to be superior (based on statistical significance analysis) and less demanding computationally than the IE filter.

The much simpler two-level white noise filter appears to have performance close to the VSD filter.

A multiple-level white or autocorrelated noise model seems to be the best compromise from the point of view of performance versus cost of implementation.

The computational requirements of the IE filter, the VSD filter, and the two-level process noise filter were, approximately, 8 : 2 : 1.

Neither of these three filters considered here managed to keep the *peak position estimation error* (during the maneuver) below the *raw position measurement RMS error* — the measurement noise standard deviation. This is a major shortcoming of most adaptive estimation schemes.

As will be shown in the next section, the multiple model approach can accomplish this (almost<sup>1</sup>) while yielding good noise reduction during the constant velocity portions of the trajectory.

<sup>1</sup>In the spirit of “Our algorithms can almost do almost everything.”

## 11.6 THE MULTIPLE MODEL APPROACH

### 11.6.1 Formulation of the Approach

In the ***multiple model (MM) approach*** it is assumed that the system obeys one of a finite number of models. Such systems are called ***hybrid***: They have both *continuous* (noise) uncertainties and *discrete* uncertainties — ***model or mode***, or ***operating regime*** uncertainties.

A ***Bayesian framework*** is used: Starting with prior probabilities of each model being correct (i.e., the system is in a particular mode), the corresponding posterior probabilities are obtained.

First the static case in which the model the system obeys is ***fixed***, that is, no switching from one mode to another occurs during the estimation process (time-invariant mode) is considered. This will result in the ***static MM estimator***. While the model that is in effect stays fixed, each model has its own dynamics, so the ***overall estimator is dynamic***.

The model, assumed to be in effect throughout the entire process, is one of  $r$  possible models (the system is in one of  $r$  modes)

$$M \in \{M_j\}_{j=1}^r \quad (11.6.1-1)$$

The prior probability that  $M_j$  is correct (the system is in mode  $j$ ) is

$$P\{M_j|Z^0\} = \mu_j(0) \quad j = 1, \dots, r \quad (11.6.1-2)$$

where  $Z^0$  is the prior information and

$$\sum_{j=1}^r \mu_j(0) = 1 \quad (11.6.1-3)$$

since the correct model is among the assumed  $r$  possible models.

It will be assumed that all models are linear-Gaussian. This approach can be used for nonlinear systems as well via linearization.

Subsequently, the dynamic situation of ***switching models*** or ***mode jumping*** is considered. In the latter case, the system undergoes transitions from one mode to another. The resulting estimator is a ***dynamic MM estimator***.

### 11.6.2 The Static Multiple Model Estimator

The ***static MM estimator*** — for fixed models — is obtained as follows.

Using Bayes' formula, the posterior probability of model  $j$  being correct, given the measurement data up to  $k$ , is given by the recursion

$$\begin{aligned} \mu_j(k) &\triangleq P\{M_j|Z^k\} = P\{M_j|z(k), Z^{k-1}\} = \frac{p[z(k)|Z^{k-1}, M_j]P\{M_j|Z^{k-1}\}}{p[z(k)|Z^{k-1}]} \\ &= \frac{p[z(k)|Z^{k-1}, M_j]P\{M_j|Z^{k-1}\}}{\sum_{i=1}^r p[z(k)|Z^{k-1}, M_i]P\{M_i|Z^{k-1}\}} \end{aligned} \quad (11.6.2-1)$$

or

$$\mu_j(k) = \frac{p[z(k)|Z^{k-1}, M_j]\mu_j(k-1)}{\sum_{i=1}^r p[z(k)|Z^{k-1}, M_i]\mu_i(k-1)} \quad j = 1, \dots, r \quad (11.6.2-2)$$

starting with the given prior probabilities (11.6.1-2).

The first term on the right-hand side above is the *likelihood function of mode*  $j$  at time  $k$ , which, under the linear-Gaussian assumptions, is given by the expression (5.2.6-7)

$$\Lambda_j(k) \triangleq p[z(k)|Z^{k-1}, M_j] = p[\nu_j(k)] = \mathcal{N}[\nu_j(k); 0, S_j(k)] \quad (11.6.2-3)$$

where  $\nu_j$  and  $S_j$  are the innovation and its covariance from the *mode-matched filter* corresponding to mode  $j$ . In a nonlinear and/or non-Gaussian situation the same Gaussian likelihood functions are used, even though they are clearly approximations.

Thus a Kalman filter matched to each mode is set up yielding *mode-conditioned state estimates* and the associated *mode-conditioned covariances*. The probability of each mode being correct — the *mode estimates* — is obtained according to (11.6.2-2) based on its likelihood function (11.6.2-3) relative to the other filters' likelihood functions. In a nonlinear situation the filters are EKF instead of KF.

This modular estimator, which is a *bank of filters*, is illustrated in Fig. 11.6.2-1.

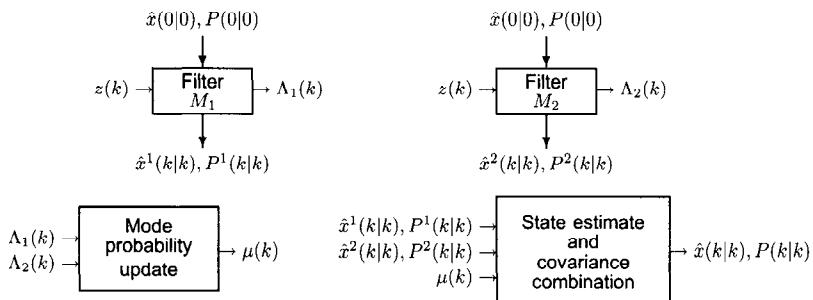


Figure 11.6.2-1: The static multiple model estimator for  $r = 2$  fixed models.

The output of each mode-matched filter is the *mode-conditioned state estimate*  $\hat{x}^j$ , the associated covariance  $P^j$  and the *mode likelihood function*  $\Lambda_j$ .

After the filters are initialized, they run recursively *on their own estimates*. Their likelihood functions are used to update the mode probabilities. The latest mode probabilities are used to combine the mode-conditioned estimates and covariances.

Under the above assumptions the pdf of the state of the system is a Gaussian mixture with  $r$  terms

$$p[x(k)|Z^k] = \sum_{j=1}^r \mu_j(k) \mathcal{N}[x(k); \hat{x}^j(k|k), P^j(k|k)] \quad (11.6.2-4)$$

The combination of the mode-conditioned estimates is done therefore as follows

$$\hat{x}(k|k) = \sum_{j=1}^r \mu_j(k) \hat{x}^j(k|k) \quad (11.6.2-5)$$

and the covariance of the combined estimate is (see Subsection 1.4.16)

$$P(k|k) = \sum_{j=1}^r \mu_j(k) \{ P^j(k|k) + [\hat{x}^j(k|k) - \hat{x}(k|k)][\hat{x}^j(k|k) - \hat{x}(k|k)]' \} \quad (11.6.2-6)$$

where the last term above is the *spread of the means* term.

The above is exact under the following assumptions:

1. The correct model is among the set of models considered,
2. The same model has been in effect from the initial time.

Assumption 1 can be considered a reasonable approximation; however, 2 is obviously not true if a maneuver has started at some time within the interval  $[1, k]$ , in which case a model change — *mode jump* — occurred.

### Convergence of the Mode Estimates

If the mode set includes the correct one and no mode jump occurs, then the probability of the true mode will converge to unity, that is, this approach yields consistent estimates of the system parameters. Otherwise the probability of the model “nearest” to the correct one will converge to unity (this is discussed in detail in [Baram78]).

### Ad Hoc Modifications for the Case of Switching Models

The following ad hoc modification can be made to the static MM estimator for estimating the state in the case of switching models: An artificial lower bound is imposed on the model probabilities (with a suitable renormalization of the remaining probabilities).

A shortcoming of the static MM estimator when used with switching models is that, in spite of the above ad hoc modification, the mismatched filters’ errors can grow to unacceptable levels. Thus, reinitialization of the filters that are mismatched is, in general, needed. This is accomplished by using the estimate from filter corresponding to the best matched model in the other filters.

It should be pointed out that the above “fixes” are automatically (and rigorously) built into the dynamic MM estimation algorithms to be discussed next.

### 11.6.3 The Dynamic Multiple Model Estimator

In this case the mode the system is in can undergo switching in time. The system is modeled by the equations

$$x(k) = F[M(k)]x(k-1) + v[k-1, M(k)] \quad (11.6.3-1)$$

$$z(k) = H[M(k)]x(k) + w[k, M(k)] \quad (11.6.3-2)$$

where  $M(k)$  denotes the mode or model “at time  $k$ ” — in effect *during the sampling period ending at  $k$* . Such systems are also called **jump-linear systems**. The mode jump process is assumed **left-continuous** (i.e., the impact of the new model starts at  $t_k^+$ ).

The mode at time  $k$  is assumed to be among the possible  $r$  modes

$$M(k) \in \{M_j\}_{j=1}^r \quad (11.6.3-3)$$

The continuous-valued vector  $x(k)$  and the discrete variable  $M(k)$  are sometimes referred to as the **base state** and the **modal state**, respectively.

For example,

$$F[M_j] = F_j \quad (11.6.3-4)$$

$$v(k-1, M_j) \sim \mathcal{N}(u_j, Q_j) \quad (11.6.3-5)$$

that is, the structure of the system and/or the statistics of the noises might be different from model to model. The mean  $u_j$  of the noise can model a maneuver as a deterministic input.

The  $l$ th **mode history** — or **sequence of models** — through time  $k$  is denoted as

$$M^{k,l} = \{M_{i_{1,l}}, \dots, M_{i_{k,l}}\} \quad l = 1, \dots, r^k \quad (11.6.3-6)$$

where  $i_{\kappa,l}$  is the model index at time  $\kappa$  from history  $l$  and

$$1 \leq i_{\kappa,l} \leq r \quad \kappa = 1, \dots, k \quad (11.6.3-7)$$

Note that the number of histories increases *exponentially with time*.

For example, with  $r = 2$  one has at time  $k = 2$  the following  $r^k = 4$  possible sequences (histories) as shown below:

$l$	$i_{1,l}$	$i_{2,l}$
1	1	1
2	1	2
3	2	1
4	2	2

It will be assumed that the **mode (model) switching** — that is, the **mode jump process** — is a Markov process (Markov chain) with known mode transition probabilities

$$p_{ij} \triangleq P\{M(k) = M_j | M(k-1) = M_i\} \quad (11.6.3-8)$$

These **mode transition probabilities** will be assumed time-invariant and independent of the base state. In other words, this is a **homogeneous Markov chain**.

The system (11.6.3-1), (11.6.3-2), and (11.6.3-8) is a generalized version of a **hidden Markov model**.

The event that model  $j$  is in effect at time  $k$  is denoted as

$$M_j(k) \triangleq \{M(k) = M_j\} \quad (11.6.3-9)$$

The conditional probability of the  $l$ th history

$$\mu^{k,l} \triangleq P\{M^{k,l}|Z^k\} \quad (11.6.3-10)$$

will be evaluated next.

The  $l$ th sequence of models through time  $k$  can be written as

$$M^{k,l} = \{M^{k-1,s}, M_j(k)\} \quad (11.6.3-11)$$

where sequence  $s$  through  $k-1$  is its *parent sequence* and  $M_j$  is its last element.

Then, in view of the Markov property,

$$P\{M_j(k)|M^{k-1,s}\} = P\{M_j(k)|M_i(k-1)\} \triangleq p_{ij} \quad (11.6.3-12)$$

where  $i = s_{k-1}$ , the index of the last model in the parent sequence  $s$  through  $k-1$ .

The conditional pdf of the state at  $k$  is obtained using the total probability theorem with respect to the mutually exclusive and exhaustive set of events (11.6.3-6), as a **Gaussian mixture** with an *exponentially increasing number of terms*

$$p[x(k)|Z^k] = \sum_{l=1}^{r^k} p[x(k)|M^{k,l}, Z^k] P\{M^{k,l}|Z^k\}$$

(11.6.3-13)

Since *to each mode sequence one has to match a filter*, it can be seen that an exponentially increasing number of filters are needed to estimate the (base) state, which makes the optimal approach impractical.

The probability of a mode history is obtained using Bayes' formula as

$$\begin{aligned}
 \mu^{k,l} &= P\{M^{k,l}|Z^k\} \\
 &= P\{M^{k,l}|z(k), Z^{k-1}\} \\
 &= \frac{1}{c} p[z(k)|M^{k,l}, Z^{k-1}] P\{M^{k,l}|Z^{k-1}\} \\
 &= \frac{1}{c} p[z(k)|M^{k,l}, Z^{k-1}] P\{M_j(k), M^{k-1,s}|Z^{k-1}\} \\
 &= \frac{1}{c} p[z(k)|M^{k,l}, Z^{k-1}] P\{M_j(k)|M^{k-1,s}, Z^{k-1}\} \mu^{k-1,s} \\
 &= \frac{1}{c} p[z(k)|M^{k,l}, Z^{k-1}] P\{M_j(k)|M^{k-1,s}\} \mu^{k-1,s} \quad (11.6.3-14)
 \end{aligned}$$

where  $c$  is the normalization constant.

If the current mode depends only on the previous one (i.e., it is a Markov chain), then

$$\mu^{k,l} = \frac{1}{c} p[z(k)|M^{k,l}, Z^{k-1}]P\{M_j(k)|M_i(k-1)\}\mu^{k-1,s} \quad (11.6.3-15)$$

or

$$\boxed{\mu^{k,l} = \frac{1}{c} p[z(k)|M^{k,l}, Z^{k-1}]p_{ij}\mu^{k-1,s}} \quad (11.6.3-16)$$

where  $i = s_{k-1}$  is the last model of the parent sequence  $s$ .

The above equation shows that *conditioning on the entire past history* is needed even if the random parameters are Markov.

## Practical Algorithms

The only way to avoid the exponentially increasing number of histories, which have to be accounted for, is by going to suboptimal techniques.

A simple-minded suboptimal technique is to keep, say, the  $N$  histories with the largest probabilities, discard the rest, and renormalize the probabilities such that they sum up to unity.

The **generalized pseudo-Bayesian (GPB)** approaches combine histories of models that differ in “older” models. The first-order GPB, denoted as GPB1, considers only the possible models in the last sampling period. The second-order version, GPB2, considers all the possible models in the last two sampling periods. These algorithms require  $r$  and  $r^2$  filters to operate in parallel, respectively.

Finally, the **interacting multiple model (IMM)** estimation algorithm will be presented. This algorithm is conceptually similar to GPB2, but requires only  $r$  filters to operate in parallel.

## The Mode Transition Probabilities

The **mode transition probabilities** (11.6.3-8), indicated as assumed to be known, are actually **estimator design parameters** to be selected in the design process of the algorithm. This will be discussed in detail in Subsections 11.6.7 and 11.7.3.

## Note

The GPB1 and IMM algorithms have approximately the same computational requirements as the static (fixed model) MM algorithm, but do not require ad hoc modifications as the latter, which is actually obsolete for switching models.

### 11.6.4 The GPB1 Multiple Model Estimator for Switching Models

In the **generalized pseudo-Bayesian estimator of first order (GPB1)**, at time  $k$  the state estimate is computed under each possible current model — a total of  $r$  possibilities (hypotheses) are considered. All histories that differ in “older” models are combined together.

The total probability theorem is thus used as follows:

$$\begin{aligned} p[x(k)|Z^k] &= \sum_{j=1}^r p[x(k)|M_j(k), Z^k] P\{M_j(k)|Z^k\} \\ &= \sum_{j=1}^r p[x(k)|M_j(k), z(k), Z^{k-1}] \mu_j(k) \\ &\approx \sum_{j=1}^r p[x(k)|M_j(k), z(k), \hat{x}(k-1|k-1), P(k-1|k-1)] \mu_j(k) \end{aligned} \quad (11.6.4-1)$$

Thus at time  $k-1$  there is a single ***lumped estimate***  $\hat{x}(k-1|k-1)$  and the associated covariance that summarize (approximately) the past  $Z^{k-1}$ . From this, one carries out the prediction to time  $k$  and the update at time  $k$  under  $r$  hypotheses, namely,

$$\hat{x}^j(k|k) = \hat{x}[k|k; M_j(k), \hat{x}(k-1|k-1), P(k-1|k-1)] \quad j = 1, \dots, r \quad (11.6.4-2)$$

$$P^j(k|k) = P[k|k; M_j(k), P(k-1|k-1)] \quad j = 1, \dots, r \quad (11.6.4-3)$$

After the update, the estimates are combined with the weightings  $\mu_j(k)$  (detailed later), resulting in the new combined estimate  $\hat{x}(k|k)$ . In other words, *the  $r$  hypotheses are merged into a single hypothesis at the end of each cycle*.

Finally, the mode (or model) probabilities are updated.

Figure 11.6.4-1 describes this estimator, which requires  $r$  filters in parallel.

The output of each model-matched filter is the **mode-conditioned state estimate**  $\hat{x}^j$ , the associated covariance  $P^j$  and the **mode likelihood function**  $\Lambda_j$ .

After the filters are initialized, they run recursively using *the previous combined estimate*. Their likelihood functions are used to update the mode probabilities. The latest mode probabilities are used to combine the model-conditional estimates and covariances.

The structure of this algorithm is

$$(N_e; N_f) = (1; r) \quad (11.6.4-4)$$

where  $N_e$  is the **number of estimates** at the start of the cycle of the algorithm and  $N_f$  is the **number of filters** in the algorithm.

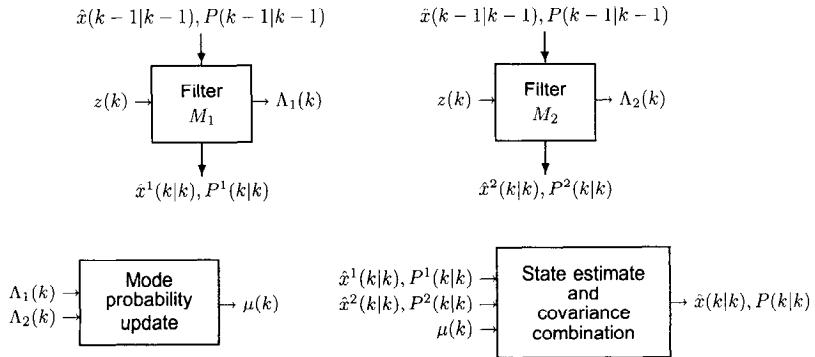


Figure 11.6.4-1: The GPB1 MM estimator for  $r = 2$  switching models (one cycle).

## The Algorithm

One cycle of the algorithm consists of the following:

1. **Mode-matched filtering** ( $j = 1, \dots, r$ ). Starting with  $\hat{x}(k-1|k-1)$ , one computes  $\hat{x}^j(k|k)$  and the associated covariance  $P^j(k|k)$  through a filter matched to  $M_j(k)$ . The likelihood functions

$$\Lambda_j(k) = p[z(k)|M_j(k), Z^{k-1}] \quad (11.6.4-5)$$

corresponding to these  $r$  filters are evaluated as

$$\boxed{\Lambda_j(k) = p[z(k)|M_j(k), \hat{x}(k-1|k-1), P(k-1|k-1)]} \quad (11.6.4-6)$$

2. **Mode probability update** ( $j = 1, \dots, r$ ). This is done as follows:

$$\begin{aligned} \mu_j(k) &\triangleq P\{M_j(k)|Z^k\} \\ &= P\{M_j(k)|z(k), Z^{k-1}\} \\ &= \frac{1}{c} p[z(k)|M_j(k), Z^{k-1}] P\{M_j(k)|Z^{k-1}\} \\ &= \frac{1}{c} \Lambda_j(k) \sum_{i=1}^r P\{M_j(k)|M_i(k-1), Z^{k-1}\} \\ &\quad \cdot P\{M_i(k-1)|Z^{k-1}\} \end{aligned} \quad (11.6.4-7)$$

which yields with  $p_{ij}$  the known *mode transition probabilities*,

$$\boxed{\mu_j(k) = \frac{1}{c} \Lambda_j(k) \sum_{i=1}^r p_{ij} \mu_i(k-1)} \quad (11.6.4-8)$$

where  $c$  is the normalization constant

$$c = \sum_{j=1}^r \Lambda_j(k) \sum_{i=1}^r p_{ij} \mu_i(k-1) \quad (11.6.4-9)$$

**3. State estimate and covariance combination.** The latest combined state estimate and covariance are obtained according to the summation (11.6.4-1) as

$$\boxed{\hat{x}(k|k) = \sum_{j=1}^r \hat{x}^j(k|k) \mu_j(k)} \quad (11.6.4-10)$$

$$\boxed{P(k|k) = \sum_{j=1}^r \mu_j(k) \{ P^j(k|k) + [\hat{x}^j(k|k) - \hat{x}(k|k)][\hat{x}^j(k|k) - \hat{x}(k|k)]' \}} \quad (11.6.4-11)$$

### 11.6.5 The GPB2 Multiple Model Estimator for Switching Models

In the **generalized pseudo-Bayesian estimator of second order** (or **GPB2**), at time  $k$  the state estimate is computed under *each possible current and previous model* — a total of  $r^2$  hypotheses (histories) are considered. All histories that differ only in “older” models are merged.

The total probability theorem is thus used as follows:

$$\begin{aligned} p[x(k)|Z^k] &= \sum_{j=1}^r \sum_{i=1}^r p[x(k)|M_j(k), M_i(k-1), Z^k] P\{M_i(k-1)|M_j(k), Z^k\} \\ &\quad \cdot P\{M_j(k)|Z^k\} \\ &= \sum_{j=1}^r \sum_{i=1}^r p[x(k)|M_j(k), z(k), M_i(k-1), Z^{k-1}] \mu_{i|j}(k-1|k) \mu_j(k) \\ &\approx \sum_{j=1}^r \sum_{i=1}^r p[x(k)|M_j(k), z(k), \hat{x}^i(k-1|k-1), P^i(k-1|k-1)] \\ &\quad \cdot \mu_{i|j}(k-1|k) \mu_j(k) \end{aligned} \quad (11.6.5-1)$$

that is, the past  $\{M_i(k-1), Z^{k-1}\}$  is approximated by the **mode-conditioned estimate**  $\hat{x}^i(k-1|k-1)$  and associated covariance.

Thus at time  $k-1$  there are  $r$  estimates and covariances, each to be predicted to time  $k$  and updated at time  $k$  under  $r$  hypotheses, namely,

$$\hat{x}^{ij}(k|k) \triangleq \hat{x}[k|k; M_j(k), \hat{x}^i(k-1|k-1), P^i(k-1|k-1)] \quad i, j = 1, \dots, r \quad (11.6.5-2)$$

$$P^{ij}(k|k) \triangleq P[k|k; M_j(k), P^i(k-1|k-1)] \quad i, j = 1, \dots, r \quad (11.6.5-3)$$

After the update, the estimates corresponding to the same latest model hypothesis are combined with the weightings  $\mu_{i|j}(k-1|k)$ , detailed later, resulting

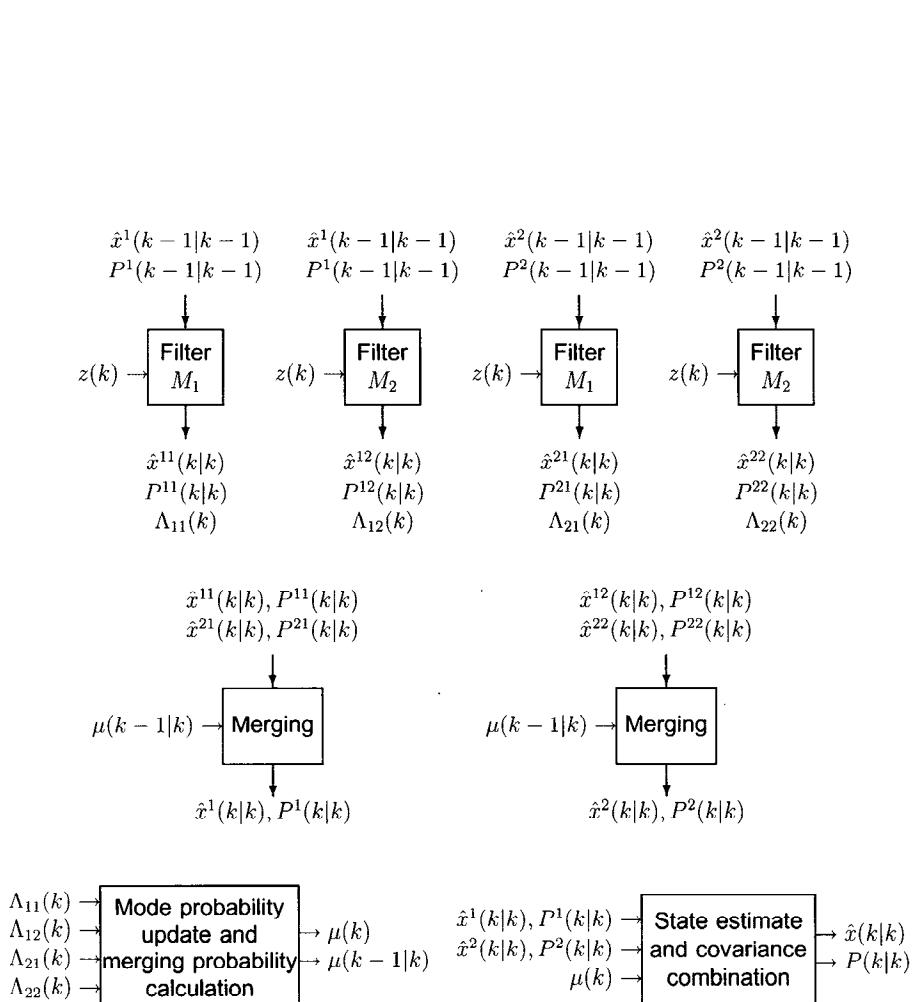


Figure 11.6.5-1: The GPB2 MM estimator for  $r = 2$  models (one cycle).

in  $r$  estimates  $\hat{x}^j(k|k)$ . In other words, the  $r^2$  hypotheses are *merged* into  $r$  at the end of each estimation cycle.

Figure 11.6.5-1 describes this algorithm, which requires  $r^2$  parallel filters. The structure of the GPB2 algorithm is

$$(N_e; N_f) = (r; r^2) \quad (11.6.5-4)$$

where  $N_c$  is the *number of estimates* at the start of the cycle of the algorithm and  $N_f$  is the *number of filters* in the algorithm.

## The Algorithm

One cycle of the algorithm consists of the following:

1. **Mode-matched filtering** ( $i, j = 1, \dots, r$ ). Starting with  $\hat{x}^i(k-1|k-1)$ , one computes  $\hat{x}^{ij}(k|k)$  and the associated covariance  $P^{ij}(k|k)$  through a filter matched to  $M_j(k)$ . The likelihood functions corresponding to these  $r^2$  filters

$$\Lambda_{ij}(k) = p[z(k)|M_j(k), M_i(k-1), Z^{k-1}] \quad (11.6.5-5)$$

are evaluated as

$$\boxed{\Lambda_{ij}(k) = p[z(k)|M_j(k), \hat{x}^i(k-1|k-1), P^i(k-1|k-1)] \quad i, j = 1, \dots, r} \quad (11.6.5-6)$$

2. **Calculation of the merging probabilities** ( $i, j = 1, \dots, r$ ). The probability that mode  $i$  was in effect at  $k-1$  if mode  $j$  is in effect at  $k$  is, conditioned on  $Z^k$ ,

$$\begin{aligned} \mu_{i|j}(k-1|k) &\triangleq P\{M_i(k-1)|M_j(k), Z^k\} \\ &= P\{M_i(k-1)|z(k), M_j(k), Z^{k-1}\} \\ &= \frac{1}{c_j} P[z(k), M_j(k)|M_i(k-1), Z^{k-1}] P\{M_i(k-1)|Z^{k-1}\} \\ &= \frac{1}{c_j} p[z(k)|M_j(k), M_i(k-1), Z^{k-1}] \\ &\quad \cdot P\{M_j(k)|M_i(k-1), Z^{k-1}\} P\{M_i(k-1)|Z^{k-1}\} \end{aligned} \quad (11.6.5-7)$$

where  $P[\cdot]$  denotes a mixed pdf-probability. Thus the *merging probabilities* are

$$\boxed{\mu_{i|j}(k-1|k) = \frac{1}{c_j} \Lambda_{ij}(k) p_{ij} \mu_i(k-1) \quad i, j = 1, \dots, r} \quad (11.6.5-8)$$

where

$$c_j = \sum_{i=1}^r \Lambda_{ij}(k) p_{ij} \mu_i(k-1) \quad (11.6.5-9)$$

The **mode transition probabilities**  $p_{ij}$  are assumed to be known — their selection is part of the algorithm design process.

3. **Merging** ( $j = 1, \dots, r$ ). The state estimate corresponding to  $M_j(k)$  is obtained by combining the estimates (11.6.5-2) according to the inner summation in (11.6.5-1) as follows

$$\hat{x}^j(k|k) = \sum_{i=1}^r \hat{x}^{ij}(k|k) \mu_{ij}(k-1|k) \quad j = 1, \dots, r \quad (11.6.5-10)$$

The covariance corresponding to the above is

$$P^j(k|k) = \sum_{i=1}^r \mu_{ij}(k-1|k) \{ P^{ij}(k|k) \\ + [\hat{x}^{ij}(k|k) - \hat{x}^j(k|k)][\hat{x}^{ij}(k|k) - \hat{x}^j(k|k)]' \} \quad (11.6.5-11)$$

4. **Mode probability updating** ( $j = 1, \dots, r$ ). This is done as follows

$$\begin{aligned} \mu_j(k) &\triangleq P\{M_j(k)|z(k), Z^{k-1}\} \\ &= \frac{1}{c} P[z(k), M_j(k)|Z^{k-1}] \\ &= \frac{1}{c} \sum_{i=1}^r P[z(k), M_j(k)|M_i(k-1), Z^{k-1}] P\{M_i(k-1)|Z^{k-1}\} \\ &= \frac{1}{c} \sum_{i=1}^r p(z(k)|M_j(k), M_i(k-1), Z^{k-1}) \\ &\quad \cdot P\{M_j(k)|M_i(k-1), Z^{k-1}\} \mu_i(k-1) \end{aligned} \quad (11.6.5-12)$$

Thus the updated **mode probabilities** are

$$\mu_j(k) = \frac{1}{c} \sum_{i=1}^r \Lambda_{ij}(k) p_{ij} \mu_i(k-1) = \frac{c_j}{c} \quad j = 1, \dots, r \quad (11.6.5-13)$$

where  $c_j$  is the expression from (11.6.5-9) and  $c$  is the normalization constant

$$c = \sum_{j=1}^r c_j \quad (11.6.5-14)$$

5. **State estimate and covariance combination**. The latest state estimate and covariance for output only are

$$\hat{x}(k|k) = \sum_{j=1}^r \hat{x}^j(k|k) \mu_j(k) \quad (11.6.5-15)$$

$$P(k|k) = \sum_{j=1}^r \mu_j(k) \{ P^j(k|k) + [\hat{x}^j(k|k) - \hat{x}(k|k)][\hat{x}^j(k|k) - \hat{x}(k|k)]' \} \quad (11.6.5-16)$$

### 11.6.6 The Interacting Multiple Model Estimator

In the *interacting multiple model (IMM) estimator*, at time  $k$  the state estimate is computed under *each possible current model* using  $r$  filters, with each filter using a different combination of the previous model-conditioned estimates — *mixed initial condition*.

The total probability theorem is used as follows to yield  $r$  filters running in parallel:

$$\begin{aligned} p[x(k)|Z^k] &= \sum_{j=1}^r p[x(k)|M_j(k), Z^k] P\{M_j(k)|Z^k\} \\ &= \sum_{j=1}^r p[x(k)|M_j(k), z(k), Z^{k-1}] \mu_j(k) \end{aligned} \quad (11.6.6-1)$$

The model-conditioned posterior pdf of the state, given by

$$p[x(k)|M_j(k), z(k), Z^{k-1}] = \frac{p[z(k)|M_j(k), x(k)]}{p[z(k)|M_j(k), Z^{k-1}]} p[x(k)|M_j(k), Z^{k-1}] \quad (11.6.6-2)$$

reflects one cycle of the state estimation filter matched to model  $M_j(k)$  starting with the prior, which is the last term above.

The total probability theorem is now applied to the last term above (the prior), yielding

$$\begin{aligned} p[x(k)|M_j(k), Z^{k-1}] &= \sum_{i=1}^r p[x(k)|M_j(k), M_i(k-1), Z^{k-1}] \\ &\quad \cdot P\{M_i(k-1)|M_j(k), Z^{k-1}\} \\ &\approx \sum_{i=1}^r p[x(k)|M_j(k), M_i(k-1), \{\hat{x}^l(k-1|k-1), P^l(k-1|k-1)\}_{l=1}^r] \\ &\quad \cdot \mu_{i|j}(k-1|k-1) \\ &= \sum_{i=1}^r p[x(k)|M_j(k), M_i(k-1), \hat{x}^i(k-1|k-1), P^i(k-1|k-1)] \\ &\quad \cdot \mu_{i|j}(k-1|k-1) \end{aligned} \quad (11.6.6-3)$$

The second line above reflects the approximation that the past through  $k-1$  is summarized by  $r$  model-conditioned estimates and covariances. The last line of (11.6.6-3) is a mixture with weightings, denoted as  $\mu_{i|j}(k-1|k-1)$ , different for each current model  $M_j(k)$ . This mixture is assumed to be a mixture of Gaussian pdfs (a Gaussian sum) and then approximated via moment matching

by a single Gaussian (details given later):

$$\begin{aligned}
 p[x(k)|M_j(k), Z^{k-1}] &= \sum_{i=1}^r \mathcal{N}[x(k); E[x(k)|M_j(k), \hat{x}^i(k-1|k-1)], \text{cov}[\cdot]] \\
 &\quad \cdot \mu_{i|j}(k-1|k-1) \\
 &\approx \mathcal{N}\left[x(k); \sum_{i=1}^r E[x(k)|M_j(k), \hat{x}^i(k-1|k-1)] \mu_{i|j}(k-1|k-1), \text{cov}[\cdot]\right] \\
 &= \mathcal{N}\left[x(k); E[x(k)|M_j(k), \sum_{i=1}^r \hat{x}^i(k-1|k-1) \mu_{i|j}(k-1|k-1)], \text{cov}[\cdot]\right]
 \end{aligned} \tag{11.6.4-4}$$

The last line above follows from the linearity of the Kalman filter and amounts to the following:

The input to the filter matched to model  $j$  is obtained from an **interaction** of the  $r$  filters, which consists of the **mixing** of the estimates  $\hat{x}^i(k-1|k-1)$  with the weightings (probabilities)  $\mu_{i|j}(k-1|k-1)$ , called the **mixing probabilities**.

The above is equivalent to hypothesis merging *at the beginning* of each estimation cycle [Blom88]. More specifically, the  $r$  hypotheses, instead of “fanning out” into  $r^2$  hypotheses (as in the GPB2 — see Fig. 11.6.5-1), are “mixed” into a new set of  $r$  hypotheses as shown in Fig. 11.6.6-1. This is the key feature that yields  $r$  hypotheses with  $r$  filters, rather than with  $r^2$  filters as in the GPB2 algorithm.

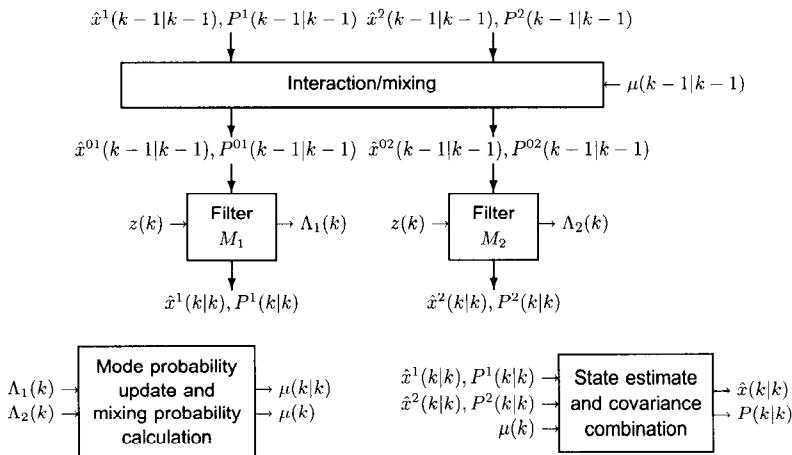


Figure 11.6.6-1: The IMM estimator (one cycle).

Figure 11.6.6-1 describes this algorithm, which consists of  $r$  interacting filters operating in parallel. The mixing is done at the input of the filters with

the probabilities, detailed later in (11.6.6-6), conditioned on  $Z^{k-1}$ . In contrast to this, the GPB2 algorithm has  $r^2$  filters and a somewhat similar mixing is done, but at their outputs, with the probabilities (11.6.5-7), conditioned on  $Z^k$ .

The structure of the IMM algorithm is

$$(N_e; N_f) = (r; r) \quad (11.6.6-5)$$

where  $N_e$  is the *number of estimates* at the start of the cycle of the algorithm and  $N_f$  is the *number of filters* in the algorithm.

## The Algorithm

One cycle of the algorithm consists of the following:

1. **Calculation of the mixing probabilities** ( $i, j = 1, \dots, r$ ). The probability that mode  $M_i$  was in effect at  $k-1$  given that  $M_j$  is in effect at  $k$  conditioned on  $Z^{k-1}$  is

$$\begin{aligned} \mu_{i|j}(k-1|k-1) &\triangleq P\{M_i(k-1)|M_j(k), Z^{k-1}\} \\ &= \frac{1}{\bar{c}_j} P\{M_j(k)|M_i(k-1), Z^{k-1}\} P\{M_i(k-1)|Z^{k-1}\} \end{aligned} \quad (11.6.6-6)$$

The above are the **mixing probabilities**, which can be written as

$$\mu_{i|j}(k-1|k-1) = \frac{1}{\bar{c}_j} p_{ij} \mu_i(k-1) \quad i, j = 1, \dots, r \quad (11.6.6-7)$$

where the normalizing constants are

$$\bar{c}_j = \sum_{i=1}^r p_{ij} \mu_i(k-1) \quad j = 1, \dots, r \quad (11.6.6-8)$$

Note the difference between (11.6.6-6), where the conditioning is  $Z^{k-1}$ , and (11.6.5-7), where the conditioning is  $Z^k$ . This is what makes it possible to carry out the mixing at the *beginning* of the cycle, rather than the standard merging at the *end* of the cycle.

2. **Mixing** ( $j = 1, \dots, r$ ). Starting with  $\hat{x}^i(k-1|k-1)$ , one computes the mixed initial condition for the filter matched to  $M_j(k)$  according to (11.6.6-4) as

$$\hat{x}^{0j}(k-1|k-1) = \sum_{i=1}^r \hat{x}^i(k-1|k-1) \mu_{i|j}(k-1|k-1) \quad j = 1, \dots, r \quad (11.6.6-9)$$

The covariance corresponding to the above is

$$\boxed{P^{0j}(k-1|k-1) = \sum_{i=1}^r \mu_{ij}(k-1|k-1) \left\{ P^i(k-1|k-1) + [x^i(k-1|k-1) - \hat{x}^{0j}(k-1|k-1)] \cdot [x^i(k-1|k-1) - \hat{x}^{0j}(k-1|k-1)]^T \right\}} \quad j = 1, \dots, r \quad (11.6.6-10)}$$

3. **Mode-matched filtering** ( $j = 1, \dots, r$ ). The estimate (11.6.6-9) and covariance (11.6.6-10) are used as input to the filter matched to  $M_j(k)$ , which uses  $z(k)$  to yield  $\hat{x}^j(k|k)$  and  $P^j(k|k)$ .

The likelihood functions corresponding to the  $r$  filters

$$\Lambda_j(k) = p[z(k)|M_j(k), Z^{k-1}] \quad (11.6.6-11)$$

are computed using the mixed initial condition (11.6.6-9) and the associated covariance (11.6.6-10) as

$$\Lambda_j(k) = p[z(k)|M_j(k), \hat{x}^{0j}(k-1|k-1), P^{0j}(k-1|k-1)] \quad j = 1, \dots, r \quad (11.6.6-12)$$

that is,

$$\boxed{\Lambda_j(k) = \mathcal{N}[z(k); \hat{z}^j[k|k-1; \hat{x}^{0j}(k-1|k-1)], S^j[k; P^{0j}(k-1|k-1)]]} \quad j = 1, \dots, r \quad (11.6.6-13)$$

4. **Mode probability update** ( $j = 1, \dots, r$ ). This is done as follows:

$$\begin{aligned} \mu_j(k) &\triangleq P\{M_j(k)|Z^k\} \\ &= \frac{1}{c} p[z(k)|M_j(k), Z^{k-1}] P\{M_j(k)|Z^{k-1}\} \\ &= \frac{1}{c} \Lambda_j(k) \sum_{i=1}^r P\{M_j(k)|M_i(k-1), Z^{k-1}\} P\{M_i(k-1)|Z^{k-1}\} \\ &= \frac{1}{c} \Lambda_j(k) \sum_{i=1}^r p_{ij} \mu_i(k-1) \quad j = 1, \dots, r \end{aligned} \quad (11.6.6-14)$$

or

$$\boxed{\mu_j(k) = \frac{1}{c} \Lambda_j(k) \bar{c}_j} \quad j = 1, \dots, r \quad (11.6.6-15)$$

where  $\bar{c}_j$  is the expression from (11.6.6-8) and

$$c = \sum_{j=1}^r \Lambda_j(k) \bar{c}_j \quad (11.6.6-16)$$

is the normalization constant for (11.6.6-15).

**5. Estimate and covariance combination.** Combination of the model-conditioned estimates and covariances is done according to the mixture equations

$$\hat{x}(k|k) = \sum_{j=1}^r \hat{x}^j(k|k) \mu_j(k) \quad (11.6.6-17)$$

$$P(k|k) = \sum_{j=1}^r \mu_j(k) \{ P^j(k|k) + [\hat{x}^j(k|k) - \hat{x}(k|k)][\hat{x}^j(k|k) - \hat{x}(k|k)]' \} \quad (11.6.6-18)$$

This combination is *only* for output purposes — it is not part of the algorithm recursions.

### Note

One possible generalization of the IMM estimator is the “second-order IMM” with an extra period depth. While the derivations are rather lengthy, it has been reported that this algorithm is *identical* to the GPB2 [Barret90].

## 11.6.7 An Example with the IMM Estimator

The use of the IMM estimator is illustrated on the example simulated in Section 11.5 where several of the earlier techniques were compared. The results presented in the sequel, which were obtained with DynaEst™, deal with the turn of 90° over 20 sampling periods.

A two-model IMM, designated as IMM2, was first used. This algorithm consisted of

1. A constant velocity model (second-order, with no process noise); and
2. A Wiener process acceleration model (third-order model) with process noise (acceleration increment over a sampling period)  $q = 10^{-3} = (0.0316 \text{ m/s}^2)^2$ .

Note that the acceleration in this case ( $0.075 \text{ m/s}^2$ ) corresponds to about  $2.4\sqrt{q}$ .

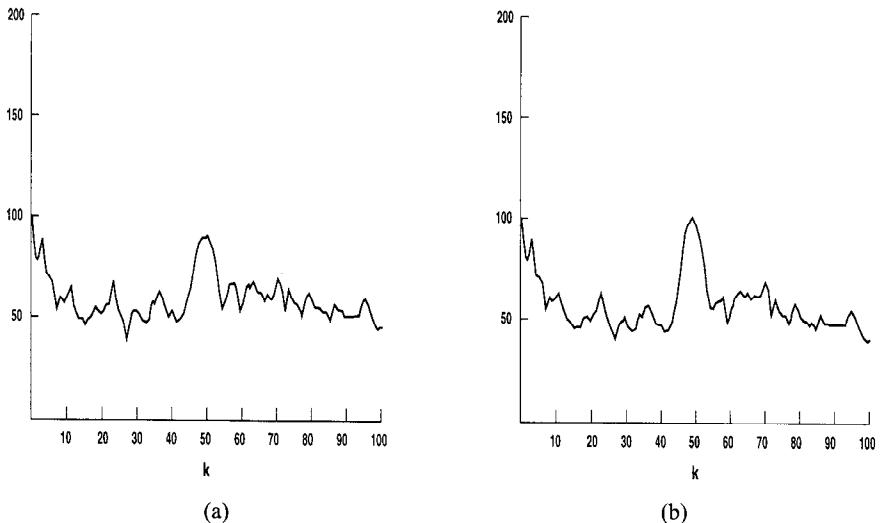
The Markov chain transition matrix between these models was taken as

$$[p_{ij}] = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix} \quad (11.6.7-1)$$

The final results were not very sensitive to these values (e.g.,  $p_{11}$  can be between 0.8 and 0.98). The lower (higher) value will yield less (more) peak error during maneuver but higher (lower) RMS error during the quiescent period; that is, it has a higher (lower) bandwidth.

A three-model IMM, designated as IMM3, was also used. This consisted of the above two models plus another one:

3. A constant acceleration (third-order) model without process noise.



**Figure 11.6.7-1:** Position RMS error. (a) IMM2, (b) IMM3.

The Markov chain transition matrix was

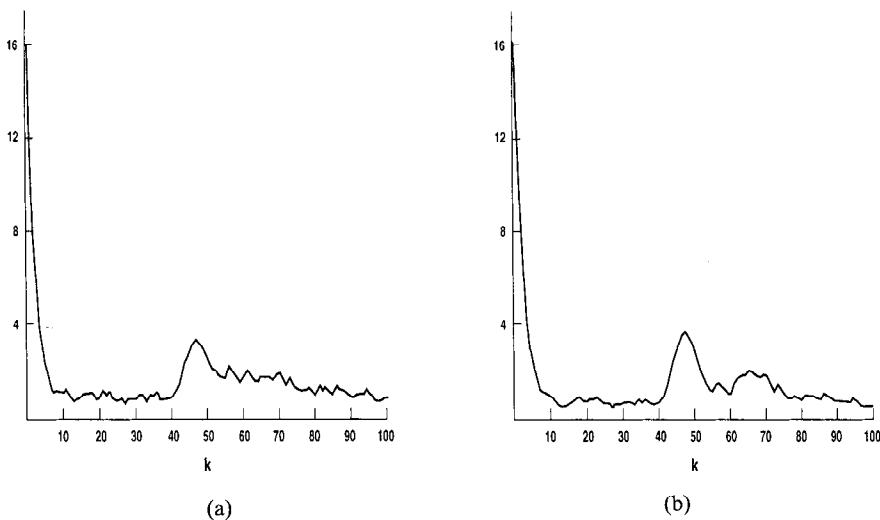
$$[p_{ij}] = \begin{bmatrix} 0.95 & 0.05 & 0 \\ 0.33 & 0.34 & 0.33 \\ 0 & 0.05 & 0.95 \end{bmatrix} \quad (11.6.7-2)$$

Figure 11.6.7-1 shows the position (coordinate  $\xi$ ) RMS error from 50 Monte Carlo runs for the IMM2 and IMM3, with the estimator design parameters as indicated above. Comparing with Fig. 11.5.3-1a, it can be seen that the peak errors are approximately equal to the measurement noise standard deviation (which is 100 m), that is, substantially smaller than with the IE or VSD. During the nonaccelerating period the errors are somewhat larger, but still there is an **RMS noise reduction factor** of 2; this corresponds to a **variance noise reduction factor** of 4.

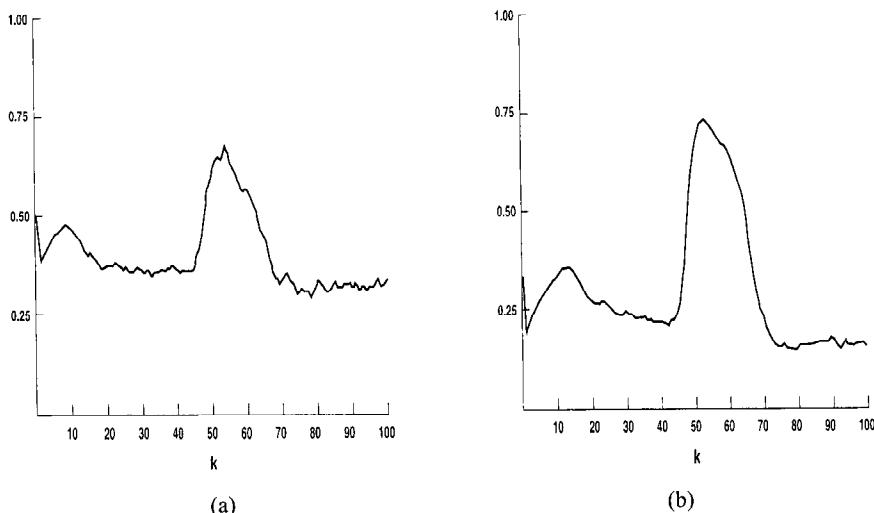
Similar results for the velocity (in coordinate  $\xi$ ) can be observed by comparing Figs. 11.6.7-2 with Fig. 11.5.3-1b.

Overall, the IMM2 and the IMM3 perform similarly in the position estimation; the IMM3 has a somewhat better velocity estimation capability.

Figure 11.6.7-3 shows the evolution in time of the maneuvering mode probabilities in IMM2 and IMM3. The figure illustrates the **soft switching** that takes place when a maneuver is initiated: It takes a few samples in this case to “detect” the maneuver. The “detection” of a maneuver manifests itself here as a sharp increase in the probability of the maneuvering mode — mode 2 in IMM2 and mode 3 in IMM3.



**Figure 11.6.7-2:** Velocity RMS error. (a) IMM2, (b) IMM3.



**Figure 11.6.7-3:** Maneuvering mode probability. (a) IMM2, (b) IMM3.

## Remark

These results, based on [Bar-Shalom89], are not only superior to the IE technique of Section 11.3 (illustrated in Subsection 11.5.3), but also superior to the augmented version of the IE algorithm that estimates the maneuver onset time [Bogler87].

This shows that the IMM, at the cost of about three Kalman filters, is preferable to the augmented IE technique, which has a much larger implementation complexity — of 30 to 100 Kalman filters.

### 11.6.8 Use of DynaEst<sup>TM</sup> to Design an IMM Estimator

In Subsection 5.3.3 the use of the companion software DynaEst<sup>TM</sup> for specifying a simulation scenario and designing a Kalman filter for estimation was demonstrated. In this section, the design of an IMM estimator using DynaEst is illustrated. One of the IMM estimator configurations presented in Subsection 11.6.7, IMM2, which consisted of two filter modules, is used for this. Before proceeding to design the IMM estimator, DynaEst has to be used to define the simulation scenario in Section 11.5 as shown in Figs. 5.3.3-3–5.3.3-11.

Following the design of the simulation scenario, the IMM estimator design steps are as follows.

**Selection of a Tracker** The selection of a Kalman filter as the estimator was shown in Fig. 5.3.3-13. Now, the IMM estimator is selected as the tracker as shown in Fig. 11.6.8-1.

**Structure Selection of an IMM Estimator** First, the number of filter modules in the IMM estimator has to be specified. In addition, the nature of the mode transition probability matrix  $[p_{ij}]$  has to be selected. DynaEst offers a choice between using a fixed mode transition probability matrix and using an adaptive one whose elements are modified at each sampling time based on the revisit interval and the *mean sojourn time* of each mode.

The simulation scenario defined in Section 11.5 uses a constant sampling interval  $T = 10\text{ s}$ . Hence, using a fixed mode transition probability matrix, as shown in Fig. 11.6.8-2, is appropriate.

**The Mode Transition Probability Matrix** The fixed mode transition probability matrix of the IMM2 estimator, given in (11.6.7-1), is specified in DynaEst as shown in Fig. 11.6.8-3.

**Selection of an IMM Filter Module to Design** Since the IMM estimator consists of a number of filter modules, possibly with different state-space models, the parameters for them have to be specified individually for each module. The first filter module is selected for design as shown in Fig. 11.6.8-4.

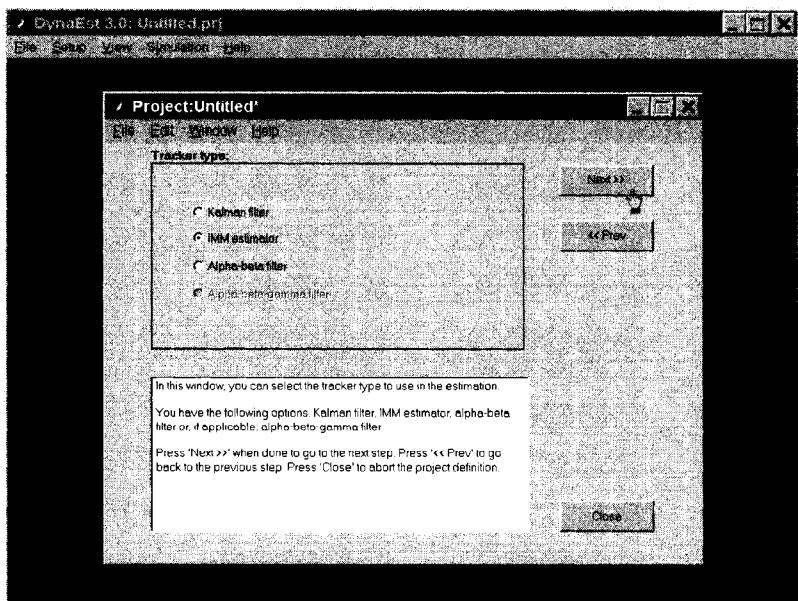


Figure 11.6.8-1: Selection of an IMM estimator as the tracker.

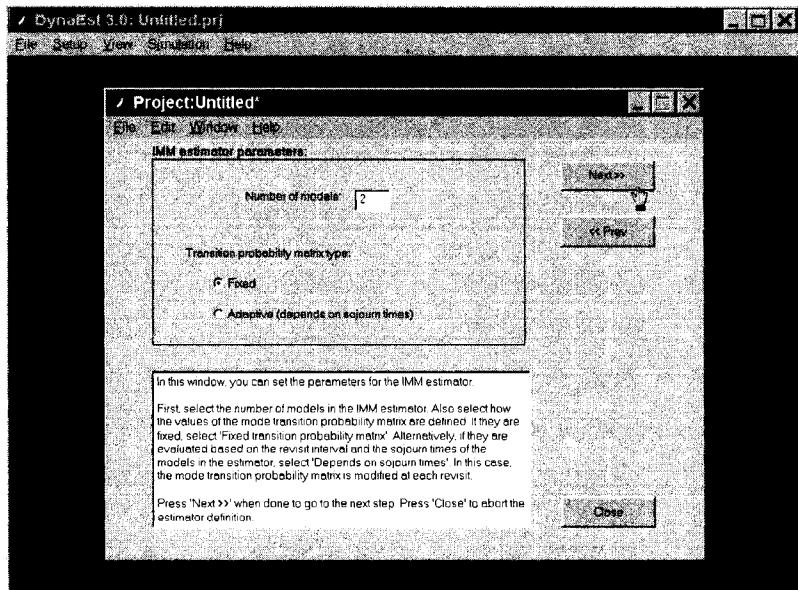
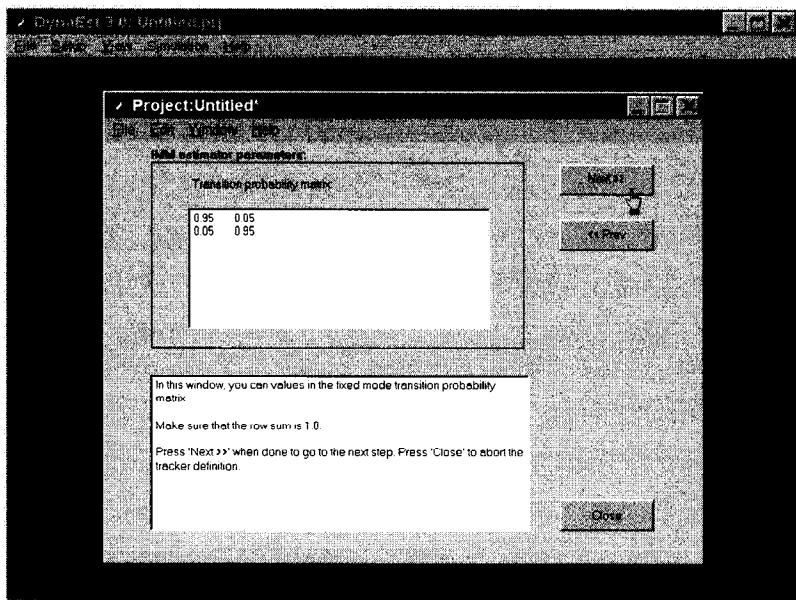
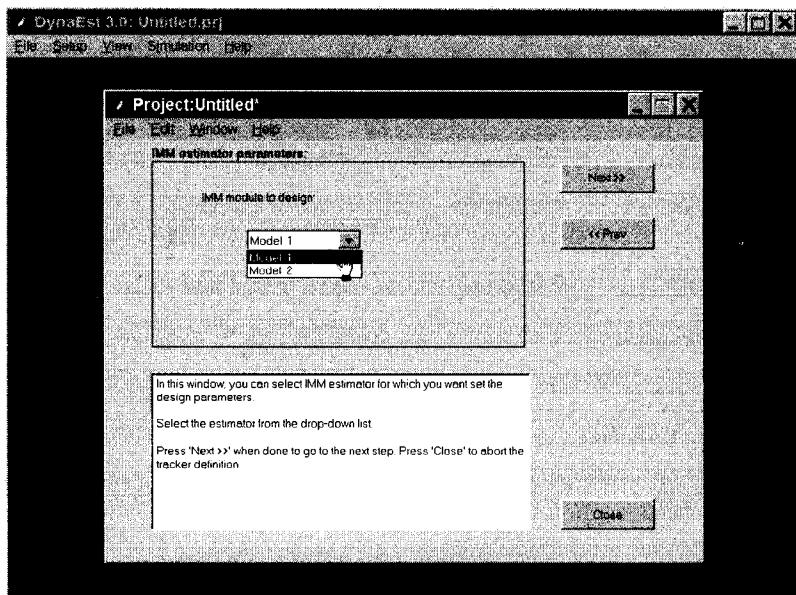


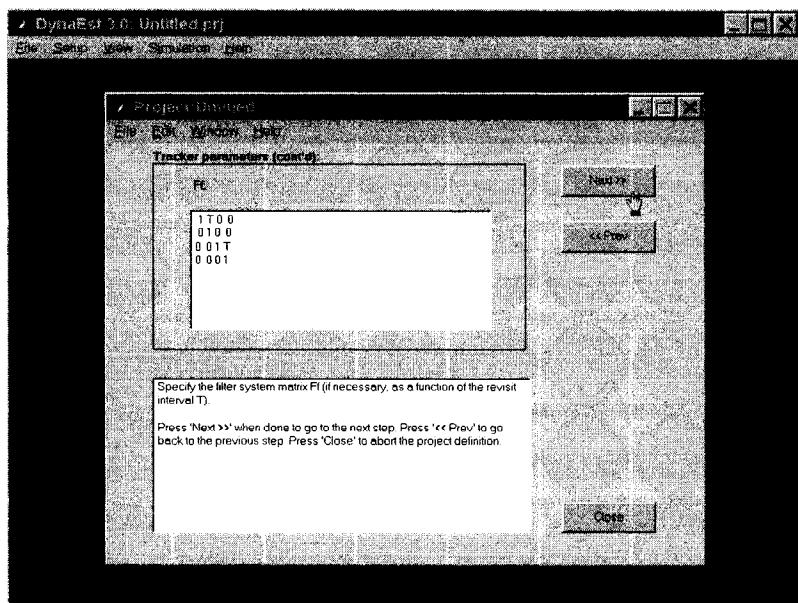
Figure 11.6.8-2: Structure selection of an IMM estimator.



**Figure 11.6.8-3:** Specification of the mode transition probability matrix.



**Figure 11.6.8-4:** Selection of an IMM filter module to design.



**Figure 11.6.8-5:** Design of a filter module.

**Design of a Filter Module** Selecting the parameters in a module is the crucial step in the IMM estimator design. This step involves selecting the assumed motion model and setting the corresponding process noise parameters in the module. DynaEst offers a sequence of windows similar to those in Figs. 5.3.3-6–5.3.3-11 for setting the motion model, noise parameters, initial estimates, and the corresponding covariance matrix. The first window in the IMM filter module design process is shown in Fig. 11.6.8-5.

**Selection of the Next Filter Module to Design** After defining the target motion model, process noise parameters, initial estimate, and the corresponding covariance matrix for the first IMM filter module, the next step is to select another filter module for designing as shown in Fig. 11.6.8-6. After selecting this filter module, its parameters are defined as described earlier.

**Viewing Estimation Results** After the IMM estimator is designed, DynaEst executes the specified number of Monte Carlo runs and offers choices to plot the estimation results as shown in Fig. 11.6.8-7. Note that the mode probability plot option is enabled only when the IMM estimator is used for estimation.

The design of the other IMM estimator, IMM3, can be carried out in a similar

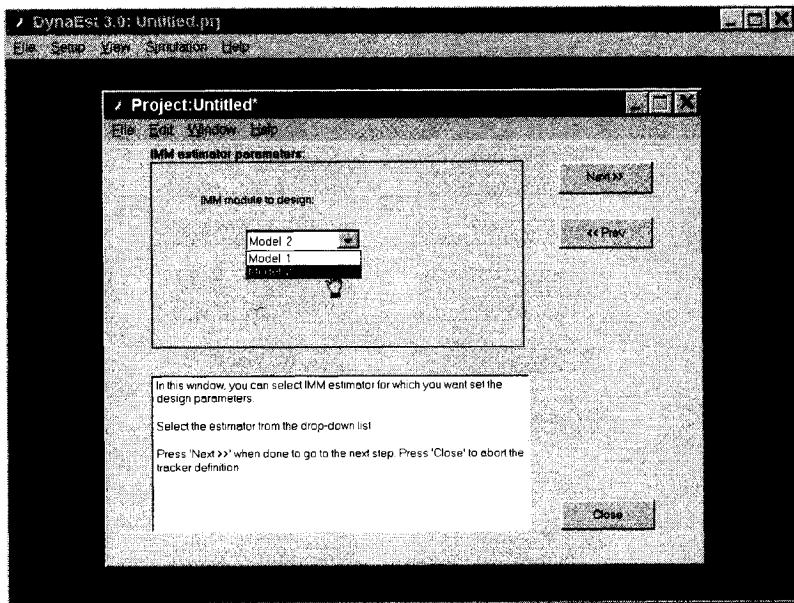


Figure 11.6.8-6: Selection of the next filter module to design.

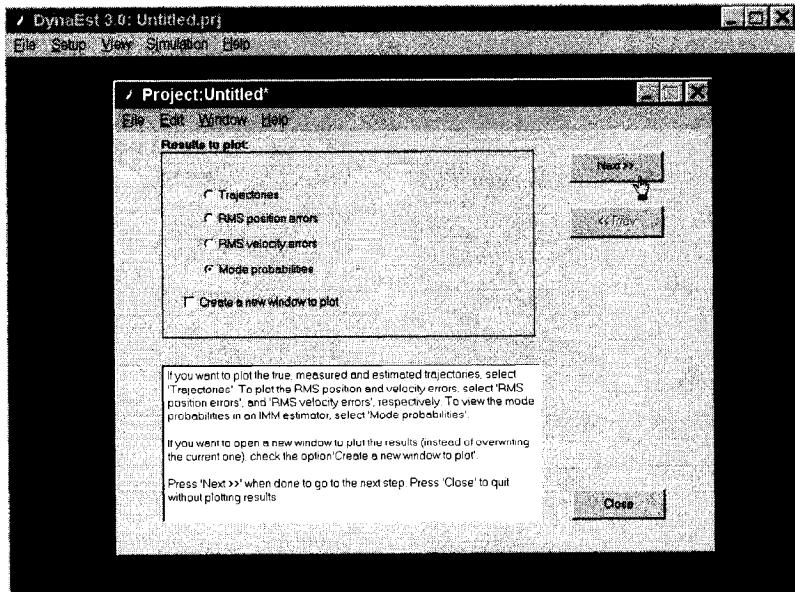


Figure 11.6.8-7: Viewing estimation results.

manner, but with the additional constant acceleration (third-order) model. This is left as an exercise to the reader.

### 11.6.9 The Multiple Model Approach — Summary

The *multiple model* or *hybrid system* approach assumes the system to be in one of a finite number of modes (i.e., that it is described by one out of a finite number of models).

Each model is characterized by its parameters — the models can differ, for instance, in the level of the process noise (its variance), a deterministic input, and/or any other parameter (different dimension state vectors are also possible).

For the *fixed model* case the estimation algorithm consists of the following:

- For each model a filter “matched” to its parameters is yielding *model-conditioned estimates and covariances*.
- A *mode probability calculator* — a *Bayesian model comparator* — updates the probability of each mode using
  - the likelihood function (innovations) of each filter
  - the prior probability of each model
- An *Estimate combiner* computes the overall estimate and the associated covariance as the weighted sum of the model-conditioned estimates and the corresponding covariance — via the (Gaussian) mixture equations.

The typical ad hoc modification of the fixed model (static) approach to handle switching models is to impose a lower bound on the probability of each model.

For systems that undergo changes in their mode during their operation — *mode jumping (model switching)* — one can obtain the *optimal multiple model estimator* which, however, consists of an exponentially increasing number of filters.

This is because the optimal approach requires conditioning on each *mode history*, and their number is increasing exponentially. Thus, suboptimal algorithms are necessary for the (realistic) mode transition situation.

The *first-order generalized pseudo-Bayesian* (GPB1) MM approach computes the state estimate accounting for each possible current model.

The *second-order generalized pseudo-Bayesian* (GPB2) MM approach computes the state estimate accounting for

- Each possible current model
- Each possible model at the previous time

The *interacting multiple model* (IMM) approach computes the state estimate that accounts for *each possible current model* using a suitable mixing of the previous model-conditioned estimates depending on the current model.

These algorithms are *decision-free* — no maneuver detection decision is needed — the algorithms undergo a *soft switching* according to the latest updated mode probabilities.

Table 11.6.9-1 presents a comparison of the complexities (in terms of the functions to be performed) of the static algorithm and the three algorithms presented for switching models.

**Table 11.6.9-1:** Comparison of complexities of the MM algorithms.

	<i>Static</i>	<i>GPB1</i>	<i>GPB2</i>	<i>IMM</i>
Number of filters	$r$	$r$	$r^2$	$r$
Number of combinations of $r$ estimates and covariances	1	1	$r + 1$	$r + 1$
Number of probability calculations	$r$	$r$	$r^2 + r$	$r^2 + r$

As can be seen from the above, the static algorithm has the same requirements as the GPB1. The IMM has only slightly higher requirements than the GPB1, but clearly significantly lower than GPB2.

In view of this, the modifications of the static algorithm for the switching situation are considered obsolete.

In view of the fact that the IMM performs significantly better than GPB1 and almost as well as GPB2 [Blom88], the IMM is considered to be the best compromise between complexity and performance.

Furthermore, the IMM has been shown to be able to keep the position estimation error not worse than the raw measurement error during the critical maneuver periods (onset and termination) and provide significant improvement (noise reduction) at other times.

## 11.7 DESIGN OF AN IMM ESTIMATOR FOR ATC TRACKING

### 11.7.1 ATC Motion Models

In *air traffic control* (ATC), civilian aircraft have two basic modes of flight:

- **Uniform motion (UM)** — the straight and level flight with a constant speed and course
- **Maneuver** — turning or climbing/descending

The horizontal and vertical motion models can be, typically, assumed to be decoupled [Wang99]. The design and evaluation of several estimators for the horizontal motion will be discussed next. The flight modes in the *horizontal plane* can be modeled by

- A nearly constant velocity model for the uniform motion, implemented as a WNA (white noise acceleration, or second-order kinematic, discussed in Section 6.5.3) model with low-level process noise
- A maneuvering model, which can be implemented as
  - a WNA model with significant process noise, commensurate with the expected maneuvers, or
  - a *nearly “coordinated turn” model*

The nearly constant velocity model is given by

$$x(k+1) = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} \frac{1}{2}T^2 & 0 \\ T & 0 \\ 0 & \frac{1}{2}T^2 \\ 0 & T \end{bmatrix} v(k) \quad (11.7.1-1)$$

where  $T$  is the sampling interval;  $x$  is the state of the aircraft, defined as

$$x = [\xi \ \dot{\xi} \ \eta \ \dot{\eta}]' \quad (11.7.1-2)$$

with  $\xi$  and  $\eta$  denoting the Cartesian coordinates of the horizontal plane; and  $v$  is a zero-mean Gaussian white noise used to model (“cover”) small accelerations, the turbulence, wind change, and so on, with an appropriate covariance  $Q$ , which is a design parameter.

The turn of a civilian aircraft usually follows a pattern known as *coordinated turn (CT)* (Subsection 4.2.2) — characterized by constant turn rate and constant speed. Although the actual turns are not exactly “coordinated” since the ground speed is the airspeed plus the wind speed, it can be suitably described by the “coordinated turn” model plus a fairly small noise representing the modeling error, resulting in the nearly coordinated turn model. The *CT model* is necessarily a nonlinear one if the turn rate is not a known constant. Augmenting the state vector (11.7.1-2) by one more component — the turn rate  $\Omega$ , that is,

$$x = [\xi \ \dot{\xi} \ \eta \ \dot{\eta} \ \Omega]' \quad (11.7.1-3)$$

the *nearly coordinated turn model* is then given by

$$x(k+1) = \begin{bmatrix} 1 & \frac{\sin \Omega(k)T}{\Omega(k)} & 0 & -\frac{1 - \cos \Omega(k)T}{\Omega(k)} & 0 \\ 0 & \cos \Omega(k)T & 0 & -\sin \Omega(k)T & 0 \\ 0 & \frac{1 - \cos \Omega(k)T}{\Omega(k)} & 1 & \frac{\sin \Omega(k)T}{\Omega(k)} & 0 \\ 0 & \sin \Omega(k)T & 0 & \cos \Omega(k)T & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} x(k)$$

$$+ \begin{bmatrix} \frac{1}{2}T^2 & 0 & 0 \\ T & 0 & 0 \\ 0 & \frac{1}{2}T^2 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix} v(k) \quad (11.7.1-4)$$

Note that the process noise  $v$  in (11.7.1-1) has different dimension from the one in (11.7.1-4).

Assuming only position measurements are available, this yields the following observation equation

$$z(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} x(k) + w(k) \quad (11.7.1-5)$$

where  $w$  is the measurement noise. Although the original radar measurements are in polar coordinates, they can be converted to Cartesian coordinates, as discussed in Subsection 10.4.3.

## 11.7.2 The EKF for the Coordinated Turn Model

Since the model in (11.7.1-4) is nonlinear, the estimation of the state (11.7.1-3) will be done via the EKF.

The equations for state prediction and the corresponding covariance, the only ones affected by the nonlinearity of (11.7.1-4), are given below. The dynamic equation (11.7.1-4) can be rewritten compactly as

$$x(k+1) = f[k, x(k)] + \Gamma_{CT}(k)v(k) \quad (11.7.2-1)$$

To obtain the predicted state  $\hat{x}(k+1|k)$ , the nonlinear function in (11.7.2-1) is expanded in Taylor series around the latest estimate  $\hat{x}(k|k)$  with terms up to first order to yield the (first-order) EKF<sup>2</sup>. The vector Taylor series expansion of (11.7.2-1) up to first order is

$$\begin{aligned} x(k+1) = & f[k, \hat{x}(k|k)] + f_x(k)[x(k) - \hat{x}(k|k)] \\ & + \text{HOT} + \Gamma_{CT}(k)v(k) \end{aligned} \quad (11.7.2-2)$$

where HOT represents the higher-order terms and

$$f_x(k) = [\nabla_x f(k, x)']' |_{x=\hat{x}(k|k)}$$

---

<sup>2</sup>The second-order EKF was found not to provide any benefit in this problem.

$$= \begin{bmatrix} 1 & \frac{\sin \hat{\Omega}(k)T}{\hat{\Omega}(k)} & 0 & -\frac{1-\cos \hat{\Omega}(k)T}{\hat{\Omega}(k)} & f_{\Omega,1}(k) \\ 0 & \cos \hat{\Omega}(k)T & 0 & -\sin \hat{\Omega}(k)T & f_{\Omega,2}(k) \\ 0 & \frac{1-\cos \hat{\Omega}(k)T}{\hat{\Omega}(k)} & 1 & \frac{\sin \hat{\Omega}(k)T}{\hat{\Omega}(k)} & f_{\Omega,3}(k) \\ 0 & \sin \hat{\Omega}(k)T & 0 & \cos \hat{\Omega}(k)T & f_{\Omega,4}(k) \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (11.7.2-3)$$

is the Jacobian of the vector  $f$  evaluated at the latest estimate of the state. The partial derivatives with respect to  $\Omega$  are given by

$$= \begin{bmatrix} f_{\Omega,1}(k) \\ f_{\Omega,2}(k) \\ f_{\Omega,3}(k) \\ f_{\Omega,4}(k) \\ \frac{(\cos \hat{\Omega}(k)T) T \hat{\dot{\xi}}(k)}{\hat{\Omega}(k)} - \frac{(\sin \hat{\Omega}(k)T) \hat{\dot{\xi}}(k)}{\hat{\Omega}(k)^2} - \frac{(\sin \hat{\Omega}(k)T) T \hat{\dot{\eta}}(k)}{\hat{\Omega}(k)} - \frac{(-1+\cos \hat{\Omega}(k)T) \hat{\dot{\eta}}(k)}{\hat{\Omega}(k)^2} \\ -(\sin \hat{\Omega}(k)T) T \hat{\dot{\xi}}(k) - (\cos \hat{\Omega}(k)T) T \hat{\dot{\eta}}(k) \\ \frac{(\sin \hat{\Omega}(k)T) T \hat{\dot{\xi}}(k)}{\hat{\Omega}(k)} - \frac{(1-\cos \hat{\Omega}(k)T) \hat{\dot{\xi}}(k)}{\hat{\Omega}(k)^2} + \frac{(\cos \hat{\Omega}(k)T) T \hat{\dot{\eta}}(k)}{\hat{\Omega}(k)} - \frac{(\sin \hat{\Omega}(k)T) \hat{\dot{\eta}}(k)}{\hat{\Omega}(k)^2} \\ (\cos \hat{\Omega}(k)T) T \hat{\dot{\xi}}(k) - (\sin \hat{\Omega}(k)T) T \hat{\dot{\eta}}(k) \end{bmatrix} \quad (11.7.2-4)$$

Based on the above expansion, the state prediction and state prediction covariance in the EKF are

$$\hat{x}(k+1|k) = f[k, \hat{x}(k|k)] \quad (11.7.2-5)$$

$$P(k+1|k) = f_x(k)P(k|k)f_x(k)' + \Gamma_{CT}(k)Q(k)\Gamma_{CT}(k)' \quad (11.7.2-6)$$

where  $Q$  is the covariance of the process noise in (11.7.2-1).

The initial estimate of the turn rate is zero since one does know whether it will be a left turn ( $\Omega > 0$ ) or<sup>3</sup> a right turn ( $\Omega < 0$ ). Since, from this initial estimate, the algorithm is capable of estimating the turn rate including its sign, there is no need for left- and right-turn models, as in some other IMM versions. Note that at the initialization step and when the estimated turn rate is zero (or when it is sufficiently close to zero), the limiting form of  $f_x(k)$ , given in

<sup>3</sup>This sign convention follows the trigonometric convention (the navigation convention has the opposite sign for the turn rates.)

(11.7.2-7), should be used, namely,

$$f_x(k)|_{\hat{\Omega}(k)=0} = \begin{bmatrix} 1 & T & 0 & 0 & -\frac{1}{2}T^2\hat{\eta}(k) \\ 0 & 1 & 0 & 0 & -T\hat{\eta}(k) \\ 0 & 0 & 1 & T & \frac{1}{2}T^2\hat{\xi}(k) \\ 0 & 0 & 0 & 1 & T\hat{\xi}(k) \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (11.7.2-7)$$

### Mixing of States of Different Dimension in the IMM

Since the dimension of the state (11.7.1-3) is 5, while that of (11.7.1-2) is 4, the mixing in the IMM estimator is done by augmenting the latter with a component that is zero. This amounts to having a zero turn rate in the constant velocity motion, which can be also seen to reduce (11.7.1-4) to (11.7.1-1). Similarly, the covariance associated with (11.7.1-2) is to be augmented by a column and a row of zeros.

### 11.7.3 Selection of Models and Parameters

To obtain the best possible results, the IMM algorithm has to be properly designed to meet the following requirements of the ATC tracking simultaneously:

- Reduce as much as possible the estimation errors during the uniform motion.
- Maintain the peak estimation error during the maneuver lower than that of the unfiltered raw measurements.
- Provide correct and timely indication of the flight mode, especially rapid detection of the maneuver.

These requirements are fulfilled by means of

- Design of aircraft motion models for all modes of flight
- Selection of the model parameters, such as the noise levels
- Determination of the parameters of the underlying Markov chain, that is, the transition probabilities

*Model selection* should consider both the quality and complexity of the model. Typically, the models used in the IMM configuration for ATC tracking will include one (a nearly constant velocity model) for the uniform motion and one or more for the maneuver — the nearly coordinated turn model of (11.7.1-4) is a typical one.

*Selection of noise levels* for each model is an important part of the estimator design. Although the uniform motion is better modeled in (11.7.1-1) with a small process noise  $v$  to model the air turbulence, winds aloft changes, and

so forth, it may be legitimate to use a somewhat larger process noise for the uniform motion to cover slow turns as well as small linear accelerations so as to ease the burden of modeling a broad range of maneuvers. The right choice of the noise level of the nearly coordinated turn model depends on what turn rate range is expected and how many models are to be used for the maneuvers.

The performance of the IMM algorithm is not very sensitive to the *choice of the transition probabilities*. However, this choice provides to a certain degree the trade-off between the peak estimation errors at the onset of the maneuver and the maximum reduction of the estimation errors during the uniform motion. The guideline for a proper choice is to match roughly the transition probabilities with the system's actual *mean sojourn time* in each mode. See problem 11-6 for the expression of the mean sojourn time in units of the sampling interval.

#### 11.7.4 The ATC Scenario

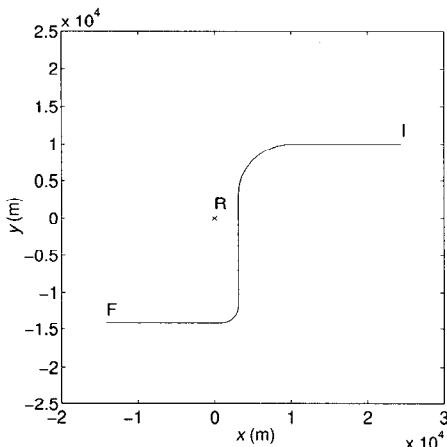
As a generic ATC tracking problem, the following scenario is considered. The radar, stationed at [0 m, 0 m], provides direct position only measurements (after the polar-to-Cartesian conversion) with RMS errors of 100 m in each of the two Cartesian coordinates.<sup>4</sup> The intervals between the samples are  $T = 5$  s. In the scenario under consideration, starting from [25,000 m, 10,000 m] at time  $t = 0$  s, the aircraft flies westward for 125 s at 120 m/s, before executing a 1°/s coordinated turn (which amounts to an acceleration of 0.2g at this speed), for 90 s. Then it flies southward for another 125 s, followed by a 3°/s turn (an acceleration of 0.6g at this speed) for 30 s. After the turn, it continues to fly westward at constant velocity. The target trajectory is shown in Fig. 11.7.4-1. This scenario leads to a maneuvering index (see Chapter 6) that is quite high (almost 1.5) and thus very little noise reduction can be achieved by a single model based state estimator, which will have to be, by necessity, conservative, i.e., designed for the maximum acceleration.

The following estimator configurations were considered:

- KF: A Kalman filter using a second order linear kinematic model (WNA) with process noise of standard deviation 1 m/s<sup>2</sup> modeling the maneuver. This is a compromise between the maximum acceleration (6 m/s<sup>2</sup>) and the constant velocity trajectory portions.
- IMM-L: An IMM estimator with two second-order linear kinematic models (WNA) with two noise levels. The one with the lower noise level with standard deviation 0.1 m/s<sup>2</sup> is used to model the uniform motion and the other one with standard deviation 2 m/s<sup>2</sup> for the maneuvers. The mode

---

<sup>4</sup>Its location under these assumptions is not relevant. In practice, the measurements are in polar coordinates and it is best to convert them into Cartesian (see Section 10.4.3). In the polar measurement case, the location of the sensor matters because the angular measurement errors are multiplied by the range from the sensor to the target.



**Figure 11.7.4-1:** The target trajectory (I — initial point, F — final point, R — radar location).

transition probability matrix  $\pi_L$  was

$$\pi_L = \begin{bmatrix} 0.95 & 0.05 \\ 0.10 & 0.90 \end{bmatrix} \quad (11.7.4-1)$$

- **IMM-CT:** An IMM estimator with one second-order kinematic model (a nearly constant velocity model with process noise standard deviation  $0.1 \text{ m/s}^2$ ) for the uniform motion and a (nearly) coordinated turn model with the turn rate (as part of the state) being estimated using an EKF. The process noise standard deviations used in the coordinated turn model were  $0.5 \text{ m/s}^2$  and  $0.2^\circ/\text{s}^2$  for the linear and turn portions of the state, respectively.

The mode transition probability matrix  $\pi_{CT}$  was

$$\pi_{CT} = \begin{bmatrix} 0.95 & 0.05 \\ 0.10 & 0.90 \end{bmatrix} \quad (11.7.4-2)$$

The initial estimates were based on one-point initializations; that is, the initial position estimate was at the first measured position and the initial velocity/turn rate was zero. The initial velocity/turn rate variances were based on their assumed maximum values. The mode set, the transition mode probability matrices, process noise standard deviations and initial estimates/variances are estimator design parameters, which should be selected based on the expected scenarios and the designer's experience.

## 11.7.5 Results and Discussion

The performances of the various estimators are shown in Table 11.7.5-1, where all the position estimation errors are for the  $\xi$  and  $\eta$  coordinates *combined*. The

*maneuver detection delay* is defined to be the latency, measured in sampling periods, from the maneuver onset time to the time that the probability of the uniform motion mode falls below 0.5. The “UM probability error” stands for the (steady-state) probability of the maneuver modes during the uniform motion. In other words, this is the estimator-calculated probability that the UM mode is not in effect while, in truth, it is.

Table 11.7.5-1: Estimation errors in the estimators (raw position measurement error 141m).

	Estimator		
	KF	IMM-L	IMM-CT
Peak position error (m)	143	138	109
UM position error (m)	123	83	71
Peak speed error (m/s)	22.5	9.3	4.8
UM speed error (m/s)	3.2	1.9	1.3
UM course error ( $^{\circ}$ )	3.8	2.7	1.8
1 $^{\circ}$ /s Maneuver detection delay (scans)	—	4	1
3 $^{\circ}$ /s Maneuver detection delay (scans)	—	3	1
UM probability error (%)	—	5.0	3.5

As presented in Table 11.7.5-1, the IMM-CT estimator yields a peak RMS position error during the mode of flight change that is almost 25% below the raw measurement error and 50% RMS error<sup>5</sup> reduction during the uniform motion. The speed estimation error during the uniform motion is 1% of the aircraft speed. The detection of the maneuver is quick: in two scans (i.e., one scan delay) for both turns in the target trajectory.

Since the coordinate-combined raw measurement error ( $1\sigma$ ) is  $100\sqrt{2} = 141$  m, the angle deviation from the straight line in one scan (sampling period) during the uniform motion due to  $1\sigma$  measurement error can be as large as

$$\tan^{-1} \frac{141 \text{ m}}{120 \text{ m/s} \cdot 5 \text{ s}} = 13.2^{\circ} \quad (11.7.5-1)$$

This is about the same as the expected course change caused by a 3 $^{\circ}$ /s turn in one scan, which is  $(3^{\circ}/\text{s}) \cdot (5 \text{ s}) = 15^{\circ}$ . In view of this, the UM probability error of only 3.5% seems quite low. The rapid detection of the maneuver, together with this small probability error, verifies the good reliability of this design in terms of providing the correct and timely information of the flight mode.

If the sampling period is shorter, one can achieve more reduction of the errors in the state estimate compared to the unfiltered radar measurements during the maneuver, and substantially more noise reduction can be achieved during the uniform motion.

<sup>5</sup>Square root of the sample average (mean) of the squared error from the Monte Carlo simulations.

Figure 11.7.5-1 presents the RMS position errors ( $\xi$  and  $\eta$  coordinates *combined*) for the IMM-L and the IMM-CT estimators as well as a baseline Kalman filter (tuned specially for this case). The initial large estimation errors are due to the fact that the initial probability of each model was set to be equal to account for the worst case of ignorance.

The following observations about the relative performances of the above estimators can be made:

IMM-CT has the peak RMS position error of about 110 m, superior to the single measurement position RMS error, which is  $100\sqrt{2} = 141$  m. Moreover, when the aircraft is not maneuvering, the RMS error is around 70 m — that is, a reduction of about 50% in the RMS error (70% in MSE). The Kalman filter has a peak RMS position error of 143 m. During the uniform motion, however, its RMS error stays at about 120 m. This estimation error of the specially tuned Kalman filter for the uniform motion can be reduced only slightly at a cost of a much higher peak error. This is the typical behavior of a single-model based Kalman filter. Between the IMM-CT estimator and the IMM-L, the former yields faster maneuver detection and lower peak estimation errors.

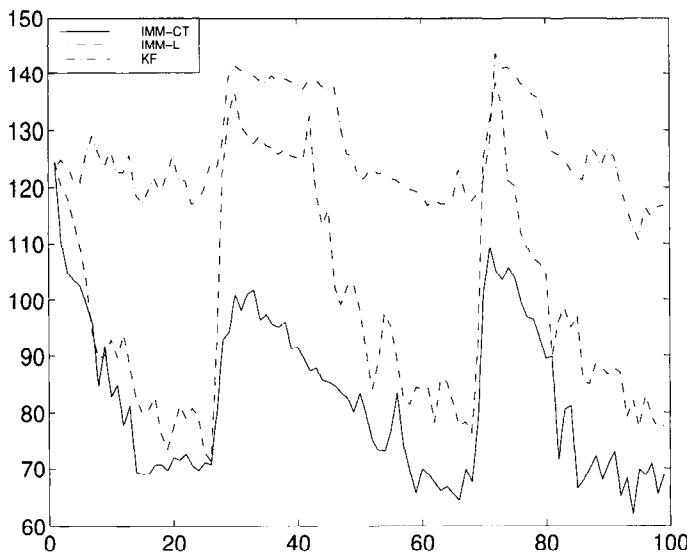
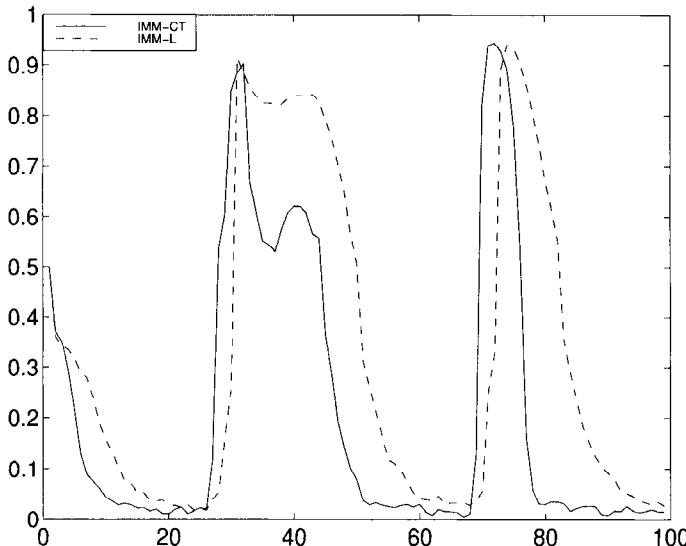


Figure 11.7.5-1: RMS position estimation errors.

Figure 11.7.5-2 shows the evolution of the average mode 2 (turning mode) probabilities in the IMM-L and IMM-CT estimators. The figure clearly indicates the rapid “detection” capability of IMM-CT — in two samples for both  $1^\circ/\text{s}$  and  $3^\circ/\text{s}$  turns. The less sophisticated IMM-L detects the turns in three to four samples.



**Figure 11.7.5-2:** Mode 2 probabilities in IMM-L and IMM-CT estimators.

### Sensitivity of the IMM to the Design Parameters

When the IMM uses several linear models with different process noise variances, a large ratio of these variances between models is needed; otherwise the models cannot be “distinguished” by the mode probability evaluator of the IMM (if they are the same, the IMM becomes a disguised KF).

Values of the maneuver onset probability between 0.05 and 0.1 yield similar results. Overall, the sensitivity of the performance to such changes in the transition probabilities are moderate.

Variable sampling interval, which is common in real systems, can be accounted for easily — see [Bar-Shalom95].

### Some Remarks

The main findings from this example are as follows:

- The KF is clearly inferior to any (reasonably designed) IMM estimator.
- The accuracy of the turn rate estimate is not very important as far as the quality of the position, speed, and course estimates are concerned. What is important is the correct and timely detection of the maneuver and the fast response of the filter to this detection.
- The IMM-CT estimator is the best choice for tracking maneuvering with its capability to track the linear as well as turn motion of the target.

When the aircraft is in uniform motion, the estimates given by the filter(s) based on the nearly coordinated turn model(s) have little effect on the overall (combined) estimate of the IMM algorithm (and on the interaction estimates for the filter based on the linear model at the next cycle) because their estimates have small weights. Consequently, while the aircraft is indeed in uniform motion, the filter matched to the uniform motion model is dominant and yields accurate estimates.

Once the aircraft starts to maneuver, the filter matched to the coordinated turn model takes over rapidly, leading to a significant reduction in the peak estimation errors. The IMM-CT estimator performs better than the KF and the IMM-L estimators during maneuvers.

A noise level switching version of the KF (with one model at a time: running a filter with a low variance process noise during the constant velocity portions of the trajectory and, following a maneuver detection, switching to a filter with a high level process noise) is not practical due to the very short duration of the maneuver. Also, in real systems where the origin of the measurements is uncertain, the “switching” technique is totally inadequate [Bar-Shalom95].

In view of the nature of maneuvers, which are not zero-mean white random processes, as the standard assumptions of the KF require, one cannot expect any of the filters to be perfectly consistent. The judgment as to which estimator design is best is a somewhat subjective one, and it should take into account the following:

- The performance (RMS errors) in both position and velocity (or speed and course): the maximum error and the average during the uniform motion
- Our desire for filter consistency
- The complexity of the implementation

Based on the above consideration, the IMM-CT is the best choice.

## 11.8 WHEN IS AN IMM ESTIMATOR NEEDED?

In Section 11.7.4 the design of an IMM estimator for a generic ATC problem was presented and it was shown that the IMM estimator performed significantly better than a Kalman filter. Specifically, the use of an IMM estimator with two filter modules, namely, a white-noise acceleration model and a coordinated turn model, resulted in about 50% reduction in RMS position errors vs. the Kalman filter. This improvement is due to the adaptive bandwidth (or the dynamic range) capability of the IMM estimator, that is, its ability to handle uniform motion (in which case it acts as a narrowband lowpass filter), as well as high-frequency maneuver motion (in which case it acts as wideband lowpass filter).

The above results raise an important question: When is an adaptive estimator — in particular, the IMM estimator — necessary? Intuitively, the higher the uncertainty — that is, the higher the maneuverability of the target — the more

a versatile tracker like the IMM estimator is needed. In the sequel we present an answer to the above question using simulation-based estimation results.

It is shown below that the choice between a (single model) Kalman filter and a multiple model tracker like the IMM estimator depends on the maneuvering index  $\lambda$  of the target, defined in (6.5.3-14). That is, it is shown that above a certain maneuvering index threshold, the performance of the IMM estimator, in terms of estimation errors, is significantly better than that of a Kalman filter, and therefore the former's use is recommended above that particular maneuvering index.

### 11.8.1 Kalman Filter vs. IMM Estimator

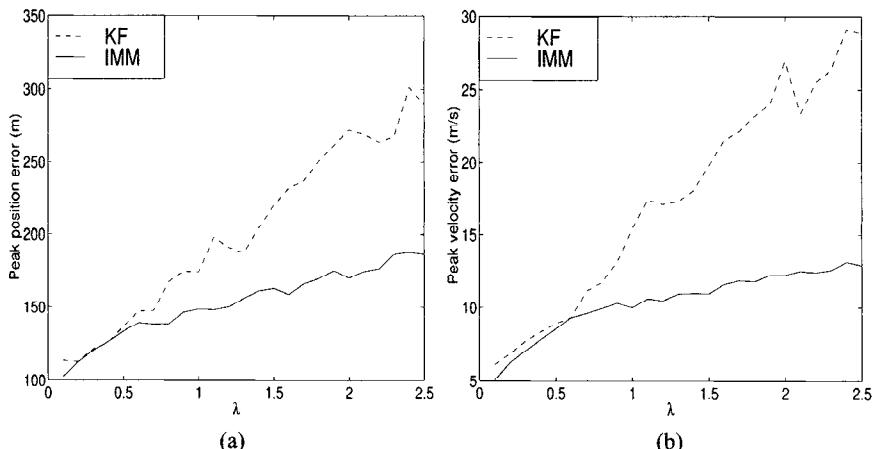
The test scenario, which is similar to the one used in Section 11.7.4, is constructed as follows:

1. The sensor measures the two-dimensional Cartesian positions with RMS errors of  $\sigma_w = 100$  m in each coordinate.
2. The revisit interval of the sensor is fixed at  $T = 5$  s.
3. Initially, the target moves at nearly a constant velocity for 60 s with process noise standard deviation  $\sigma_v = 0.2$  m/s<sup>2</sup>, which yields a very small maneuvering index of  $\lambda = 0.05$  in each coordinate.
4. Following this, the target executes a series of three maneuvers, each of which
  - lasts for 60 s
  - has a given (fixed) maneuvering index  $\lambda$  (with the corresponding white process noise)
  - is interleaved with nonmaneuvering periods lasting for 60s with  $\lambda = 0.05$ .

The level of maneuvering is varied from  $\lambda = 0.1$  to 2.5, in increments of 0.1. That is, the process noise standard deviation ranges from  $\sigma_v = 0.4$  m/s<sup>2</sup> to 10 m/s<sup>2</sup> for the above values of  $\sigma_w$  and  $T$ . The performance of the trackers based on Kalman filter and the IMM estimator are evaluated from 100 Monte Carlo runs for each of these maneuvering index values.

For the Kalman filter, a second order (WNA) linear kinematic model with  $\sigma_v$  corresponding to 0.8 of the maximum process noise standard deviation (during the maneuvering intervals) was used. This choice was made in order to retain as much as possible the ability to track maneuvering and nonmaneuvering intervals. Other values of  $\sigma_v$  yielded similar results.

The IMM estimator consisted of two WNA models designed as follows:



**Figure 11.8.1-1:** Peak RMS estimation errors. (a) Position errors, (b) velocity errors.

1. One model with low process noise with  $\sigma_v = 0.2 \text{ m/s}^2$ , the process noise standard deviation during the nonmaneuvering intervals.
2. One model with high process noise corresponding to the maneuver process noise standard deviation.
3. The mode transition probability matrix  $\pi$  was

$$\pi = \begin{bmatrix} 0.90 & 0.10 \\ 0.10 & 0.90 \end{bmatrix} \quad (11.8.1-1)$$

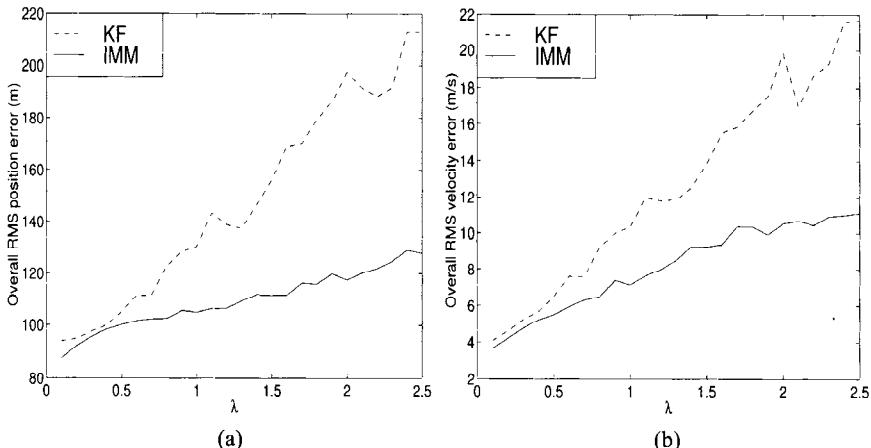
which corresponds to a sojourn time of about 60 s in each mode at  $T = 5 \text{ s}$ .

Figure 11.8.1-1 shows the peak position and velocity errors obtained with the two estimators. It can be seen that both estimators result in nearly the same peak estimation errors up to  $\lambda = 0.5$  while beyond that value of  $\lambda$  the IMM estimator yields significantly lower estimation errors.

Figure 11.8.1-2 shows the overall RMS position and velocity errors. Here also, one can notice that the IMM estimator results in significant error reduction over the Kalman filter for  $\lambda > 0.5$ .

Figure 11.8.1-3 shows the position and velocity RMS errors during the non-maneuvering intervals. It can be seen that the errors obtained with the IMM estimator remain almost constant, whereas those obtained with the Kalman filter increase with  $\lambda$ ; the latter is unable to provide noise reduction during non-maneuvering intervals because, in order to reduce the overall error, its parameters are tuned to the maneuvering intervals.

This is a direct result of KF's lack of adaptive bandwidth capability. In contrast, the IMM estimator is able to maintain the noise reduction during non-maneuvering intervals even with higher levels of target maneuverability — by



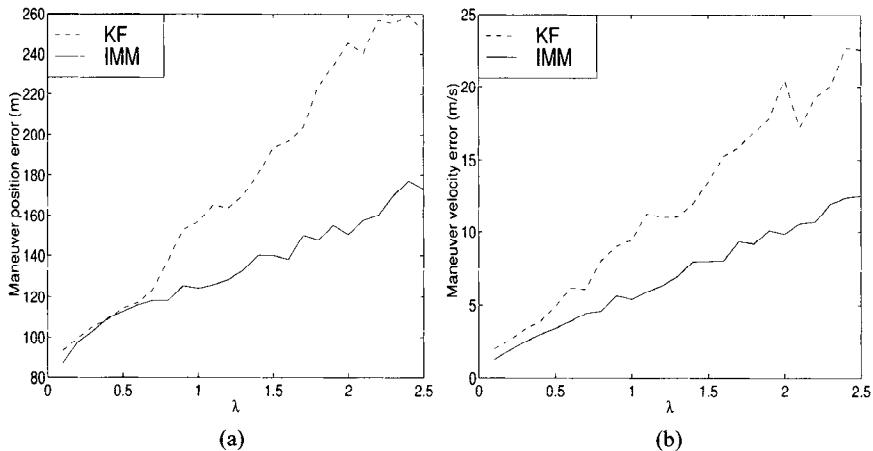
**Figure 11.8.1-2:** Overall RMS estimation errors. (a) Position errors, (b) velocity errors.

adapting the weights given to the two filter modules (mode probabilities), it gives consistently better results.

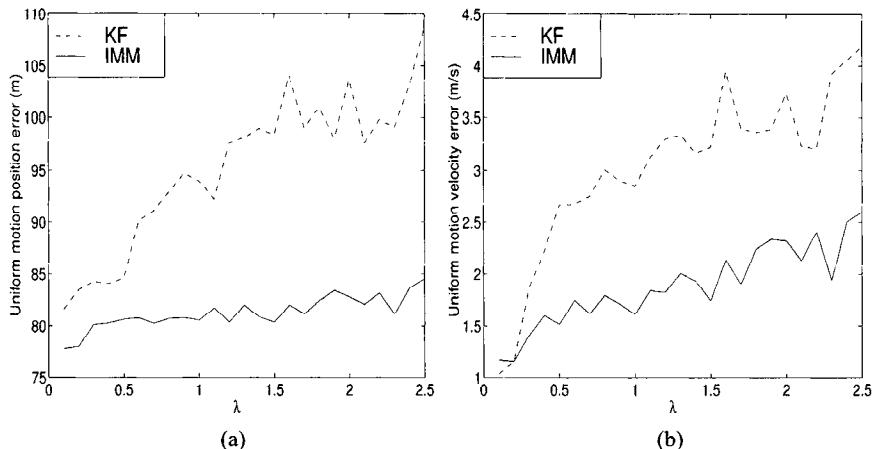
The above observation is confirmed further in Fig. 11.8.1-4, where the position and velocity RMS errors during the maneuvering intervals are shown. Notice that even with the Kalman filter's noise levels adapted to the increasing maneuverability, it is unable to match the IMM estimator's performance levels. Also notice that the breaking point for the Kalman filter is, once again, around  $\lambda = 0.5$ , up to which both estimators perform nearly the same.

These results indicate that below about  $\lambda = 0.5$ , the Kalman filter and the IMM estimator performs equally well. Other values of measurements error standard deviations and sampling intervals yielded similar results. Above this (approximate) threshold, the more sophisticated IMM estimator yields consistently better results. The **adaptive bandwidth** or **dynamic range capability** of the IMM estimator becomes a significant factor in the estimation process when the maneuverability of the target is sufficiently high. The adaptation via mode probability update helps the IMM estimator keep the estimation errors consistently low, both during maneuvers as well as during the benign motion intervals.

In summary, based on the above observations it is recommended that the **maneuvering index threshold** of  $\lambda = 0.5$  be used as a guideline in making a choice between the Kalman filter and the IMM estimator.



**Figure 11.8.1-3:** Uniform motion RMS estimation errors. (a) Position errors, (b) velocity errors.



**Figure 11.8.1-4:** Maneuver RMS estimation errors. (a) Position errors, (b) velocity errors.

## 11.9 USE OF EKF FOR SIMULTANEOUS STATE AND PARAMETER ESTIMATION

### 11.9.1 Augmentation of the State

The extended Kalman filter can be used for suboptimal state estimation in nonlinear dynamic systems.

The situation of linear systems with unknown parameters that are continuous-valued can be put in the framework of nonlinear state estimation by augmenting the base state. The **base state** is the state of the system with the parameters assumed known.

Denoting the unknown parameters as a vector  $\theta$ , the **augmented state** will be the **stacked vector** consisting of the base state  $x$  and  $\theta$

$$y(k) \triangleq \begin{bmatrix} x(k) \\ \theta \end{bmatrix} \quad (11.9.1-1)$$

The linear dynamic equation of  $x$ , with the known input  $u$  and process noise  $v$ ,

$$x(k+1) = F(\theta)x(k) + G(\theta)u(k) + v(k) \quad (11.9.1-2)$$

and the “dynamic equation” of the parameter vector (assumed time invariant)

$$\theta(k+1) = \theta(k) \quad (11.9.1-3)$$

can be rewritten as a nonlinear dynamic equation for the augmented state

$$y(k+1) = f[y(k), u(k)] + v(k) \quad (11.9.1-4)$$

The resulting nonlinear equation (11.9.1-4) is then used for an EKF to estimate the entire augmented state.

The same technique can be used if the dynamic equation of the base state is nonlinear.

The model represented by (11.9.1-3) for the parameter dynamics assumes it to be constant. Therefore, the covariance yielded by the EKF will, asymptotically, tend to zero, and consequently the filter gain for these components will tend to zero. This is because there is no process noise entering into these components of the augmented state — the **controllability condition 2** from Subsection 5.2.5, pertaining to the Riccati equation for the state estimation covariance, is not satisfied.

Since the EKF is not an optimal estimation algorithm, it will in general not yield consistent estimates for the parameters — the estimates will not converge to the true values. Thus the situation where the parameter variances tend to zero is undesirable because it will lead in practice to estimation errors much larger than the filter-calculated variances.

This can be remedied (to some extent) by assuming an *artificial process noise* (or *pseudo-noise*) entering into the parameter equation. This amounts to replacing (11.9.1-3) by

$$\theta(k+1) = \theta(k) + v_\theta(k) \quad (11.9.1-5)$$

where the parameter process noise is assumed zero mean and white. Thus the parameter is modeled as a (discrete time) Wiener process.

Any nonzero variance of this process noise will prevent the filter-calculated variances of the parameter estimates from converging to zero. Furthermore, this also gives the filter the ability to estimate *slowly varying parameters*.

The choice of the variance of the artificial process noise for the parameters — the *tuning of the filter* — can be done as follows:

1. Choose the standard deviation of the process noise as a few percent of the (estimated/guessed) value of the parameter.
2. Simulate the system and the estimator with random initial estimates (for the base state as well as the parameters) and monitor the normalized estimation errors.
3. Adjust the noise variances until, for the problem of interest, the filter is consistent — it yields estimation errors commensurate with the calculated augmented state covariance matrix. The criteria to be used are those from Section 5.4 — the estimation bias and the normalized estimation error squared (NEES).

The example of the coordinated turn motion with an unknown rate from (11.7.1-2) to (11.7.1-4) falls into this category.

## 11.9.2 An Example of Use of the EKF for Parameter Estimation

Consider the scalar system, that is, its base state  $x$  is a scalar, given by

$$x(k+1) = a(k)x(k) + b(k)u(k) + v_1(k) \quad (11.9.2-1)$$

where  $v_1(k)$  is the base state process noise and the two unknown parameters are  $a(k)$  and  $b(k)$ , possibly time-varying.

The observations are

$$z(k) = x(k) + w(k) \quad (11.9.2-2)$$

Following the procedure of the previous subsection, the augmented state is

$$y(k) \triangleq \begin{bmatrix} y_1(k) \\ y_2(k) \\ y_3(k) \end{bmatrix} \triangleq \begin{bmatrix} x(k) \\ a(k) \\ b(k) \end{bmatrix} \quad (11.9.2-3)$$

With this the nonlinear dynamic equation corresponding to (11.9.2-1) can be written as

$$y_1(k+1) = f^1[y(k), u(k)] + v_1(k) \triangleq y_1(k)y_2(k) + y_3(k)u(k) + v_1(k) \quad (11.9.2-4)$$

and the “dynamic equation” of the parameters is

$$y_i(k+1) = f^i[y(k), u(k)] + v_i(k) \triangleq y_i(k) + v_i(k) \quad i = 2, 3 \quad (11.9.2-5)$$

The augmented state equation is then

$$y(k+1) = f[y(k), u(k)] + v(k) \quad (11.9.2-6)$$

with the augmented process noise

$$v(k) \triangleq \begin{bmatrix} v_1(k) \\ v_2(k) \\ v_3(k) \end{bmatrix} \quad (11.9.2-7)$$

assumed zero mean and with covariance

$$Q = \text{diag}(q_1, q_2, q_3) \quad (11.9.2-8)$$

With the linear measurements (11.9.2-2), the only place in the EKF where linearizations are to be carried out is the base state prediction.

The second-order EKF will use the following augmented state prediction equations. From (11.9.2-4) the base state prediction can be obtained directly as

$$\hat{y}_1(k+1|k) = \hat{y}_2(k|k)\hat{y}_1(k|k) + \hat{y}_3(k|k)u(k) + P_{21}(k|k) \quad (11.9.2-9)$$

since

$$E[a(k)x(k)|Z^k] = E[y_2(k)y_1(k)|Z^k] = \hat{y}_2(k|k)\hat{y}_1(k|k) + \text{cov}[y_2(k), y_1(k)|Z^k] \quad (11.9.2-10)$$

Note that the last term in (11.9.2-9) is the only second-order term.

The predicted values of the remaining two components of the augmented state, which are the system’s unknown parameters, follow from (11.9.2-5) as

$$\hat{y}_i(k+1|k) = \hat{y}_i(k|k) \quad i = 2, 3 \quad (11.9.2-11)$$

Equations (11.9.2-9) and (11.9.2-11), in general, have to be obtained using the series expansion of the EKF, which requires evaluation of the Jacobian of the vector  $f$  and the Hessians of its components. In the present problem, the Jacobian is

$$F(k) = \begin{bmatrix} \hat{y}_2(k|k) & \hat{y}_1(k|k) & u(k) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (11.9.2-12)$$

and the Hessians are

$$f_{yy}^1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad f_{yy}^2 = 0 \quad f_{yy}^3 = 0 \quad (11.9.2-13)$$

Then, with the components of  $f$  given in (11.9.2-4) and (11.9.2-5), it can be easily shown (see problem 11-3) that using (10.3.2-4) for the augmented state, that is,

$$\hat{y}(k+1|k) = f[\hat{y}(k|k), u(k)] + \frac{1}{2} \sum_{i=1}^{n_x} e_i \text{tr}[f_{yy}^i P(k|k)] \quad (11.9.2-14)$$

yields (11.9.2-9) and (11.9.2-11).

The prediction covariance of the base state can be obtained, using (10.3.2-6) as

$$\begin{aligned} P_{11}(k+1|k) &= \hat{y}_2(k|k)^2 P_{11}(k|k) + 2\hat{y}_2(k|k)\hat{y}_1(k|k)P_{21}(k|k) \\ &\quad + 2\hat{y}_2(k|k)u(k)P_{13}(k|k) + \hat{y}_1(k|k)^2 P_{22}(k|k) + 2\hat{y}_1(k|k)u(k)P_{23}(k|k) \\ &\quad + u(k)^2 P_{33}(k|k) + P_{21}(k|k)^2 + P_{22}(k|k)P_{11}(k|k) + q_1 \end{aligned} \quad (11.9.2-15)$$

### 11.9.3 EKF for Parameter Estimation — Summary

The EKF can be used to estimate simultaneously

- the base state and
- the unknown parameters

of a system.

This is accomplished by stacking them into an augmented state and carrying out the series expansions of the EKF for this augmented state.

Since the EKF is a suboptimal technique, significant care has to be exercised to avoid filter inconsistencies. The filter has to be tuned with artificial process noise so that its estimation errors are commensurate with the calculated variances.

## 11.10 NOTES, PROBLEMS, AND TERM PROJECT

### 11.10.1 Bibliographical Notes

The “adjustment” of the process noise covariance, presented in Section 11.2, has become part of the Kalman filtering folklore. It can be found in [Jazwinski69, Jazwinski70]. Among its applications, the one presented in [Chang77] deals with state and parameter estimation for maneuvering reentry vehicles. In [Tenney77b, Tenney77a] similar ideas are used for passive tracking of maneuvering targets including reinitialization of the filter upon detection of the maneuver. In [Castella80] a continuous update of the process noise covariance is presented based on a fading memory average of the residuals. An adaptation of the filter gain based on the residuals’ deviation from orthogonality is presented in [Hampton73]. In [Springarn72] a least squares approach with exponential discount of older measurements is used for maneuvering targets. Estimation of the noise covariances has been discussed in, for example, [Li94] and [Myers76].

The switching from one model to another with different noise parameters is discussed in [McAulay73] in a manner somewhat similar to Subsection 11.2.2.

The input estimation method of Section 11.3 is based on [Chan79]. Estimation of the input and the measurement noise covariance has been discussed in [Moghaddamjoo86]. The generalized likelihood ratio technique [Willsky76b, Korn82] deals simultaneously with the estimation of the input and its onset time. Such an approach was taken by [Bogler87], who augmented the IE technique of [Chan79] to include maneuver onset time estimation; however, the resulting algorithm was very costly and performed less well than the IMM [Bar-Shalom89]. In [Demirbas87] an approach consisting of hypothesis testing via dynamic programming is proposed.

The variable dimension approach to tracking maneuvering targets discussed in Section 11.4 is from [Bar-Shalom82], which also presented the comparison of algorithms discussed in Section 11.5. An extensive comparison of various maneuver detection approaches is described in [Woolfson85].

A weighted state estimate, along the lines of Section 11.6, is presented in [Thorp73]. The multiple model (MM) approach was originally presented in [Magill65]. The Markov switching of models is discussed in [Moose73, Moose75, Moose79, Moose80, Gholson77]. The generalized pseudo-Bayesian (GPB) algorithms are proposed by [Ackerson70, Jaffer71b, Jaffer71a]. The GPB2 MM approach is based on [Chang78]. A survey of the MM techniques and their connection with failure detection is given in [Tugnait82]. Recent work in failure detection using the static MM estimator includes [Hanlon98]. The use of the IMM for failure detection, which is substantially more effective than the static MM estimator, was presented in [Zhang98]. Related work in the area of failure detection is [Willsky76a, Caglayan80, Kerr89a]; the connection between failure detection, multiobject tracking and general modeling uncertainties is discussed in [Pattipati83]. Detection-estimation algorithms that approximate the optimal algorithm of Subsection 11.6.3 are discussed in [Hadidi78, Tugnait79]. The Interacting MM algorithm is from [Blom84, Blom88]. The application of the IMM to air traffic control can be found in [Bar-Shalom98b] (Chapters 1 and 2). Section 11.7 is based on [Wang99]. Reference [Li93] discusses how one can predict the performance of an IMM algorithm without Monte Carlo simulations and illustrates this technique on examples, including a simplified ATC problem. The generalization of the IMM to second-order Markov systems was presented in [Blom85]. Smoothing with an IMM was discussed in [Helmick95]. A survey on the IMM and its numerous applications can be found in [Mazor98]. The book [Swolder99] presents a mathematical treatise on hybrid systems. A survey of MM methods can be found in [Li96].

In [Li95] a theoretical framework for evaluating the performance of hybrid estimation algorithms (with continuous and discrete uncertainties) is presented.

Adaptive sampling techniques for a maneuvering target are presented in [Blackman81]. This topic is discussed in the larger context of radar resource allocation in [Blair98, Kirubarajan98]. The use of target orientation measurements for tracking maneuvering targets is explored in [Kendrick81, Lefas84, Andrisani86, Swolder89].

## 11.10.2 Problems

**11-1 CRLB for a two-model parameter estimation problem.** Consider an estimation problem where the noise behaves according to one of two models. Using the binary random

variable  $\alpha$ , the observation can be written as

$$z = x + \alpha w_1 + (1 - \alpha) w_2$$

where  $x$  is the (unknown constant) parameter to be estimated,

$$P\{\alpha = 1\} = p_1 \quad P\{\alpha = 0\} = p_2 = 1 - p_1$$

with  $w_i \sim \mathcal{N}(0, \sigma_i^2)$  independent of each other and of  $\alpha$ . In other words, the measurement has with probability  $p_1$  accuracy  $\sigma_1$  and with probability  $p_2$  accuracy  $\sigma_2$ .

1. Write the likelihood function of  $x$ ,  $\Lambda(x) = p(z|x)$ .
2. Find  $\hat{x}^{\text{ML}}$ .
3. Evaluate the corresponding MSE.
4. Evaluate the CRLB for this estimation problem assuming  $p_1 = 0.5$ ,  $\sigma_1 = 1$ ,  $\sigma_2 = 100$ . (*Hint:* Use the fact that  $\sigma_1 \ll \sigma_2$  to approximate the integral that requires otherwise numerical evaluation.)
5. Compare the results of items 3 and 4 for the above numbers. Comment on the usefulness of the bound in this problem.

**11-2 Two-model parameter estimation problem with a prior.** Given the random parameter  $x \sim \mathcal{N}(x_0, \sigma_0^2)$  to be estimated based on the same measurement as in problem 11-1.

1. Find the MMSE estimate of  $x$  given  $z$ .
2. Find the conditional variance of  $x$  given  $z$ .

Assume  $x$ ,  $\alpha$  and  $w_i$  are all independent.

**11-3 EKF prediction equations for a system with unknown parameters.**

1. Derive the predicted augmented state from (11.9.2-14).
2. Prove (11.9.2-15) and derive the remaining terms of this covariance.

**11-4 EKF simulation for state and parameter estimation.** Consider the system (11.9.2-1) with  $x(0) = 0$ ,  $a(k) = 0.5$ ,  $b(k) = 0$ , process noise zero mean with variance  $q_1 = 0.09$ . The observations are given by (11.9.2-2) with measurement noise zero mean with variance  $r = 0.09$ . The covariance matrix of the initial augmented state estimate is  $P(0|0) = \text{diag}(0.09, 0.04)$ .

1. Implement an EKF for this problem (to estimate  $x$ ,  $a$ , and  $b$ ) without an artificial process noise. Initialize with random initial estimates according to  $P(0|0)$ . List the normalized estimation error squared (NEES) for each component of the augmented state for 100 steps.
2. Perform 100 Monte Carlo runs for the above. List the averages of the normalized estimation error for bias check and the NEES for each component for 100 steps. Indicate the corresponding 95% confidence regions.
3. If the filter needs tuning, do it and present the results. Compare also the absolute RMS estimation errors with those from (2).

**11-5 Fading memory average.** Prove (11.2.2-7).

**11-6 Sojourn time in a state for a Markov chain.** Prove

$$E[\tau_i] = \frac{1}{1 - p_{ii}}$$

where  $\tau_i$  is the expected sojourn time (in units of the sampling interval) of a Markov chain in state  $i$  (mode  $i$ ) and  $p_{ii}$  is the transition probability from state  $i$  to state  $i$ .

**11-7 Hybrid system with i.i.d. modal state sequence.**

- Find the Markov chain transition matrix that, regardless of the initial condition, yields (in one step) the fixed probability vector  $[p_1 \ p_2]'$ .
- Given that the mean sojourn time (MST) in state 1 is  $\tau_1$  periods, find, for the above case,  $\tau_2$ , the MST in state 2.

**11-8 Direct mode observation in a hybrid system.** Consider the prior pdf of  $x$  as

$$p(x) = \sum_{i=1}^n \mu_i \delta(x - x_i)$$

and the observation

$$z = x + w \quad w \sim \mathcal{N}(0, 1)$$

Find the closed-form expressions of the posterior weights  $\hat{\mu}_i$  conditioned on  $z$ .

**11-9 Fault detection.** Consider a direct discrete-time WNA motion model in one coordinate with  $T = 1$ . Let the measurement be

$$z = \begin{bmatrix} h_{11} & 0 \\ 0 & h_{22} \end{bmatrix} x + w$$

with all noises zero-mean white with unity variance.

The system in the normal mode has  $h_{11} = 1$  and  $h_{22} = 1$ . The failure modes are:

- FM1.  $h_{11} = 0$
- FM2.  $h_{22} = 0$
- FM3.  $h_{11} = 0$  and  $h_{22} = 0$ .

The system starts in the normal mode, then goes into FM1 at  $k = 10$ , recovers at  $k = 20$ , goes into FM2 at  $k = 30$ , recovers at  $k = 40$ , goes into FM3 at  $k = 50$ , and recovers at  $k = 60$ .

Let the initial condition be

- Scenario A.  $x = [0 \ 0]'$
- Scenario B.  $x = [0 \ 10]'$

both with identity matrix initial covariance.

- Design an IMM estimator for this system. Specify the design parameters and implement it with DynaEst™.
- Based on 100 runs with the same failure sequence (but independent noises and random initial state estimates), find the average number of samples it takes to detect each failure mode and recovery for the two scenarios. Define what you mean by failure mode detection. Explain the difference in performance between the two scenarios.

- 11-10 Maneuvering index threshold vs. debiasing threshold.** Show that the maneuvering index threshold  $\lambda = 0.5$  from Section 11.8 is (almost) equivalent to the debiasing threshold (10.4.3-17).

### 11.10.3 Term Project — IMM Estimator for Air Traffic Control

Design IMM estimators for the following air traffic control problem:

The Ground Truth is a target moving with a constant speed of 250 m/s with initial state in Cartesian coordinates (with position in m)

$$\boldsymbol{x} = [\xi \ \dot{\xi} \ \eta \ \dot{\eta}]' = [0 \ 0 \ 0 \ 250]'$$

The sampling period is  $T = 10$  s. At  $k = 10$  ( $t = 100$  s) it starts a left turn of  $2^\circ/\text{s}$  for 30 s, then continues straight until  $k = 20$ , at which time it turns right with  $1^\circ/\text{s}$  for 45 s, then left with  $1^\circ/\text{s}$  for 90 s, then right with  $1^\circ/\text{s}$  for 45 s, then continues straight until  $k = 50$ . (The turns are not known to the filter!)

Measurements are made starting from  $k = 0$  on the position of this target in polar coordinates (range  $r$  and azimuth  $\theta$ ) by a radar located at  $[\xi_0, \eta_0] = [-10^4, 0]$ , with

$$r = \sqrt{(\xi - \xi_0)^2 + (\eta - \eta_0)^2} \quad \theta = \tan^{-1} \left( \frac{\eta - \eta_0}{\xi - \xi_0} \right)$$

with additive white Gaussian noise with covariance  $R = \text{diag}[2500 \text{ m}^2, (1^\circ)^2]$ .

Design the following two IMM estimators:

(A) IMM-L consisting of two (linear) WNA models (Section 6.3.2), with process noise (assumed zero-mean white Gaussian) with variances  $\sigma_{v_m}^2$ ,  $m = 1, 2$ . These, together with the Markov chain transition matrix between the models, are the IMM estimator design parameters.

(B) IMM-CT consisting of one WNA model and a coordinated turn model with the turn rate as an additional state component. The CT model has an additional design parameter, the turn rate process noise variance.

The IMM estimator is initialized from the measurements at  $k = 0$  and  $k = 1$  and starts running from  $k = 2$ . Each model has initial probability 0.5 and the same initial estimate.

1. Calculate the true state according to the above specifications for  $k = 0, \dots, 50$  (the truth evolves without noise).
2. Generate the noisy measurements along the trajectory.
3. Implement the IMM estimators for this problem. Indicate the rationale for the choice of the estimator design parameters.
4. Calculate the following average results for  $N = 100$  runs for the two designs (try to have the estimated position RMS errors not to exceed the single measurement position RMS error during the maneuver — a few percent excess is OK — while having as much as possible

“noise reduction” in the straight portions of the trajectory):

$$\begin{aligned}
 \text{NORXE} &\triangleq \text{AVE}\{\tilde{x}_1(k|k)/\sqrt{P_{11}(k|k)}\} \\
 \text{FPOS} &\triangleq \sqrt{\text{AVE}\{P_{11}(k|k) + P_{33}(k|k)\}} \\
 \text{FVEL} &\triangleq \sqrt{\text{AVE}\{P_{22}(k|k) + P_{44}(k|k)\}} \\
 \text{RMSPOS} &\triangleq \text{RMS position error } (\text{Note: both coordinates combined}) \\
 \text{RMSVEL} &\triangleq \text{RMS velocity error } (\text{Note: both coordinates combined}) \\
 \text{RMSSPD} &\triangleq \text{RMS speed error } (\text{Note: speed } \triangleq \text{magnitude of velocity vector}) \\
 \text{RMSCRS} &\triangleq \text{RMS course error } (\text{Note: course } \triangleq \text{direction of velocity vector}) \\
 \text{NEES} &\triangleq \text{AVE}\{\bar{x}(k|k)'P(k|k)^{-1}\bar{x}(k|k)\} \\
 \text{MOD2PR} &\triangleq \text{AVE}\{\text{model 2 probability}\}
 \end{aligned}$$

where

$$\text{RMS}(y) \triangleq \sqrt{\frac{1}{N} \sum_{j=1}^N (y^j)^2} \quad \text{AVE}(y) \triangleq \frac{1}{N} \sum_{j=1}^N y^j$$

and  $y^j$  is the outcome of  $y$  in run  $j$ . Provide the expressions you used for calculating RMSPOS, RMSVEL, RMSSPD and RMSCRS. Indicate which is your final (“best”) design.

5. Indicate the distributions of NORXE and NEES and their 95% probability regions. Justify the final choice of the design parameters. Summarize the effect of the Markov chain transition matrix on the final results based on the designs illustrated in item 4.

6. Plot for both designs RMSPOS and FPOS; RMSVEL and FVEL; NORXE with its probability region; RMSSPD; RMSCRS; NEES with its probability region; and MODPR. Comment on the estimator bias and consistency.

7. Repeat item 4 for the best single-model filter (a standard KF) of your choice (a single design). Indicate how the design parameter was chosen and comment on its performance as compared to the best IMM.

#### DELIVERABLES:

A concise report is due.

Each student will have to make a 10- to 15-min. presentation in class. It is suggested to prepare 8–10 viewgraphs for this presentation, describing the designs and performance plots. The most important plots are:

- A single run with the true and estimated trajectories (best IMM). Plot the true trajectory as a solid line, marking the position (+) at the sampling time; the measurement ( $\square$ ) connected with a line to the corresponding true position; and the updated position ( $\circ$ ), also connected with a line to the corresponding true position at each sampling time. Provide magnified plots of the turn portions.

and the following comparison plots:

- RMSPOS for the two IMMs (RMSPOSIMM-L, RMSPOSIMM-CT) and the KF (RMSPOSKF), together, for comparison; also, indicate the (baseline) raw unfiltered position error (RMSPOSRAW).
- RMSSPD for the two IMMs and the KF (together, for comparison).
- RMSCRS for the two IMMs and the KF (together, for comparison).

- NEES for the two IMM's and the KF (together, for comparison).
- MOD2PR for the two IMM's (together, for comparison).

Develop an incremental animated display for the tracker with the following specifications:

- Plot the true position (black dot) every 0.1 s at a rate  $n$  times faster than real time, with the display speedup  $n \in [1, 10]$ . This high density sequence of points should appear as a continuous line.
- Each position measurement should appear as a box “□” connected with a red line to the true position at the corresponding time.
- The display of the measurement should be delayed from its true time by a constant  $\Delta$ , which is independent of the display speedup. Use  $\Delta = 0.1$  s to start with, then adjust to make the display more appealing if necessary.
- Each updated position should appear as bullet “●”, the starting point of a black arrow of length and direction given by the estimated velocity vector multiplied by the sampling interval  $T$ . The tip of this arrow will be then the predicted measurement for the next time under a linear motion model.
- The starting point of the arrow should be connected with a blue line to the true position at the corresponding time.
- The display of the measurement should be delayed from its true time by a constant  $2\Delta$ , which is independent of the display speedup. Use  $\Delta = 0.1$  s to start with, then adjust to make the display more appealing if necessary.
- You can use your creativity to provide sound effects (e.g., at each measurement time) to keep the audience entertained.

The display should grow incrementally, rather than being restarted at each sampling time.

# Chapter 12

## INTRODUCTION TO NAVIGATION APPLICATIONS

### 12.1 INTRODUCTION

#### 12.1.1 Navigation Applications — Outline

Navigation and target tracking are closely related.

- **Target tracking** is the estimation of the state of a moving object based on remote observations of the object.
- **Navigation** is tracking of the “own-ship” — the platform on which the sensor(s) is (are) located. It answers the question “Where am I?” by estimating one’s own state (position, velocity, attitude, and possibly acceleration).

Another problem, which incorporates both of the above, is

- **Guidance**, which is concerned with questions like “Which way leads to the destination?” It determines an appropriate sequence of actions to reach a destination. Guidance is a *control problem* based on the information obtained from, typically, a *navigation* system and, when the destination (target) is a moving object, a system that *tracks* this object.

The primary considerations for the development, choice, and design of a navigation device include cost, high accuracy, self-containment, security, worldwide capability, and high availability, among others. Certain navigation systems, such as inertial systems, are capable of being a stand-alone means for navigation mainly because they are self-contained. Other navigation devices can be used only as navigation aids to improve the navigation accuracy.

Estimation and filtering techniques, especially Kalman filtering, have found extensive application in navigation systems because they can significantly improve their accuracy.

This chapter presents an introduction to the application of estimation and filtering techniques in the Global Positioning System (GPS), inertial navigation systems, and their integration.

### 12.1.2 Navigation Applications — Summary of Objectives

Describe complementary filtering for navigation systems using multiple sensors.  
 Discuss briefly Inertial Navigation Systems (INS)

- State models
- Sensor models

Describe the Global Positioning System (GPS)

- The GPS principle
- The GPS observables and their accuracies
- State models
- Measurement models

Example of IMM vs. KF for navigation with GPS.

Discuss integrated navigation and present several schemes for fusion of the data from different sensors

- Centralized estimation fusion
- Decentralized estimation fusion
- Use of complementary filtering for integration

## 12.2 COMPLEMENTARY FILTERING FOR NAVIGATION

A key estimation and filtering concept in the processing of navigation data is the so-called *complementary filtering*. It is also one of the distinctive features of the state estimators for navigation. The philosophy of complementary filtering is best explained in the framework of Wiener filtering.

Given noisy measurements  $z(t) = s(t) + n(t)$  of a signal  $s(t)$ , the problem of Wiener filtering is to recover the signal  $s(t)$  from the measurement  $z(t)$  optimally in the sense that the mean-square error between the recovered signal and the true signal is minimized (see Section 9.5). One of the fundamental assumptions of the Wiener filtering is that the signal  $s(t)$  is a stationary random process with known power spectral density.

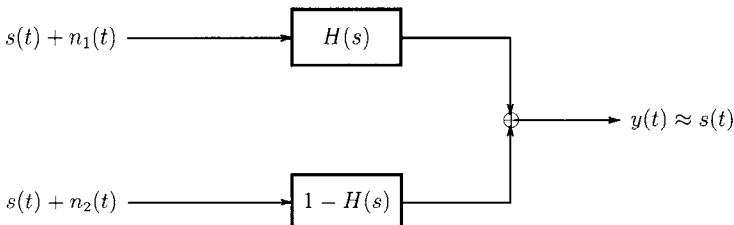
### 12.2.1 The Operation of Complementary Filtering

Consider the case where two independent measurements of the same signal are available, namely,

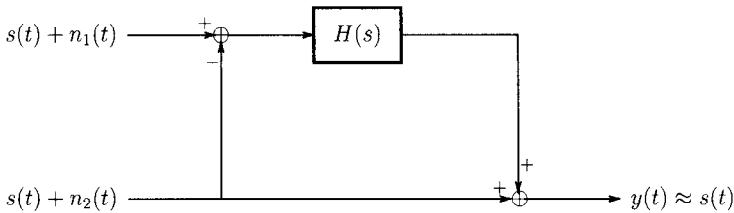
$$z_1(t) = s(t) + n_1(t) \quad (12.2.1-1)$$

$$z_2(t) = s(t) + n_2(t) \quad (12.2.1-2)$$

If the signal  $s(t)$  can be modeled well as a random process with known spectral characteristics, Wiener filtering techniques can be used, at least in



**Figure 12.2.1-1:** A form of complementary filter.



**Figure 12.2.2-1:** An alternative form of complementary filter.

principle, to recover it optimally from the two noisy measurements  $z_1(t)$  and  $z_2(t)$ .

If, however, the signal  $s(t)$  cannot be properly modeled as a random process with known spectral characteristics, Wiener filtering techniques are not applicable. Instead, a complementary filter can be used. This is illustrated in Fig. 12.2.1-1. Note that with this architecture, the *signal passes through the filter without distortion* regardless of what  $H(s)$  is, which can be designed to remove the two noises as much as possible, if they have different spectral characteristics. Specifically,  $H$  should attenuate  $n_1(t)$  and let  $n_2(t)$  pass as much as possible.

## 12.2.2 The Implementation of Complementary Filtering

Clearly, one has, from Fig. 12.2.1-1,

$$Y(s) = S(s) + N_1(s)H(s) + N_2(s)[1 - H(s)] \quad (12.2.2-1)$$

$$= S(s) + N_2(s) + [N_1(s) - N_2(s)]H(s) \quad (12.2.2-2)$$

An equivalent and more commonly used implementation of the complementary filter is thus given in Fig. 12.2.2-1.

The complementary filtering satisfies an additional constraint important for many applications: The signal is not distorted. Nevertheless, in the case where the signal  $s(t)$  can be properly modeled as a random process with known spectral characteristics, complementary filtering is a conservative philosophy. As a result, compared with Wiener filtering, its performance will suffer to some extent. However, if the signal-to-noise ratio in the measurements is high (say, 20 dB), the performance improvement of the Wiener filtering over the complementary filtering is insignificant.

Navigation signals can rarely be well-modeled as random processes, not to mention that their spectral characteristics, especially the signal-noise cross-power spectra, would not be known in practice. In addition, these signals usually have a much greater power than the noise and thus the performance degradation due to the conservativeness of the complementary filtering is often quite marginal.

Complementary filters are often used in two levels in navigation: (1) Within an inertial navigation system, the filter is used to compensate the measurement errors of the INS system; and (2) for the integration of the inertial system and other navigation systems, the filter is used to estimate the estimation errors of the primary navigation system.

### Example

The heading  $\psi$  of a vehicle is to be estimated based on measurements of a fluxgate compass and a directional (rate) gyro, respectively, given in Laplace domain by

$$\psi_c(s) = \psi(s) + \psi_n(s) \quad (12.2.2-3)$$

$$R_g(s) = R(s) + R_n(s) \quad (12.2.2-4)$$

where  $R(s) = s\psi(s)$  is the heading change rate, and  $\psi_n$  and  $R_n$  are compass and gyro measurement noise, respectively. It is known that the compass provides reliable information at low frequencies only (i.e., the spectrum of  $\psi_n$  is mostly in the high-frequency region), while in the same low-frequency band the gyro exhibits a bias and drift phenomenon and is thus useful at higher frequencies (i.e., the spectrum of  $R_n$  is mostly in the low-frequency region).

This is a typical example where complementary filtering is useful. Note first that (12.2.2-3)–(12.2.2-4) can be converted into the standard form of (12.2.1-1)–(12.2.1-2) by setting  $Z_1(s) = \psi_c(s)$  and  $Z_2(s) = R_g(s)/s$ . It then follows from (12.2.2-1) that the estimate of  $\psi$  provided by a complementary filter has the form

$$\hat{\psi}(s) = \psi(s) + \psi_n(s)H(s) + (R_n(s)/s)[1 - H(s)] \quad (12.2.2-5)$$

Thus the estimation error is

$$\tilde{\psi}(s) = \psi_n(s)H(s) + (R_n(s)/s)[1 - H(s)] \quad (12.2.2-6)$$

Since  $R_n$  and  $\psi_n$  are significant in the low- and high-frequency regions, respectively,  $H(s)$  should be low pass to attenuate  $\psi_n$  while letting  $R_n$  pass.

The simplest such filter is  $H(s) = \frac{k}{s+k}$ . The gain  $k$  is a design parameter to meet a target break frequency dictated by the frequency characteristics of the compass and gyro. Kalman filtering techniques can be used to determine the optimal  $k$  value because a complementary filter can be treated in principle as a simple Kalman filter.

In practice, this simple complementary filter structure is modified to meet additional constraints, such as to reject the steady-state bias of the rate gyro.

## 12.3 INERTIAL NAVIGATION SYSTEMS

An *inertial navigation system (INS)* is a dead-reckoning system. It provides a self-contained, nonjamming, and nonradiating means for navigation. It can be used as a stand-alone system or can be combined with some other navigation systems.

An INS has three basic functions: sensing, computing, and outputting. The sensors are *accelerometers* and *gyroscopes (gyros, for short)*, which measure linear acceleration and rotational rate (angular velocity) of the vehicle, respectively. Estimation and filtering techniques are used in the computation of the navigation solutions based on the measurements from these sensors. In order to provide three-dimensional navigation for a vehicle with six degrees of freedom, three gyros are positioned in such a way to define a three-dimensional coordinate system and a triad of accelerometers are mounted on the three orthogonal axes.

The principle of navigation by an INS system is simple. With a precise knowledge of the initial position and attitude of the vehicle, at any time, single- and double-integration of the linear accelerations give the velocity and position of the vehicle, respectively, and the attitude is obtained by integrating rotational rates from the gyro measurements. However, the presence of random errors as well as unknown constant (or nearly constant) errors (e.g., drift rates and biases) makes the navigation problem quite complex.

Inertial navigation systems are either gimbaled or strapdown. In a *gimbaled system*, also known as a *platform system*, a stabilized physical platform is maintained through a servo system on which the accelerometers and gyros are mounted and thus isolated from the rotations of the vehicle. This not only greatly reduces the requirements on the inertial sensors but also greatly simplifies the navigation computation since the attitude of the platform with respect to a navigation reference frame is preserved. However, the need to have a stabilized platform makes a gimbaled system more complicated, costly, and less reliable than a strapdown system.

In a *strapdown system*, there is no stabilized platform — the gyros and accelerometers are fixed (strapped down) to the vehicle body directly. The re-

quired reference frame is maintained mathematically by the navigation computer. Although generally less accurate than gimbaled systems, strapdown systems are employed in more and more small vehicles thanks to the recent advances in laser and optical gyros and computers.

## 12.4 MODELS FOR INERTIAL NAVIGATION SYSTEMS

The state of a state estimator model for an INS is usually chosen to be made up of various error components, such as the position, velocity, and attitude *errors* of the vehicle. In this sense, the state estimator is a *complementary filter* since its output is used to correct errors of the output of the INS. An alternative formulation is to use a full state in terms of position, velocity, and attitude per se, which, however, often requires an unreasonably high sampling rate. The error-state formulation has the advantage that any filter failure will not lead to the breakdown of the entire navigation system, in addition to its improved accuracy for nonlinear system dynamics inherent of an INS.

### 12.4.1 State Models

A simplified linear model for the errors as state components of a (horizontal) single-axis navigation system is

$$\begin{aligned}\ddot{x} &= \beta - g\phi \\ \dot{\phi} &= \dot{x}/R_e + \varepsilon\end{aligned}\tag{12.4.1-1}$$

where

- $x, \dot{x}$  = position and velocity errors, respectively
- $\phi$  = platform tilt error
- $\beta$  = accelerometer bias
- $\varepsilon$  = gyro (angular) drift rate
- $g$  = acceleration of gravity
- $R_e$  = radius of the earth

This is known as a Schuler loop (see, e.g., [Brown97]). This third order system has a pair of complex poles with a natural frequency known as the Schuler frequency.

### 12.4.2 Sensor Error Models

The gyro drift rate  $\varepsilon$  is often modeled as the sum  $\varepsilon = \varepsilon_1 + \varepsilon_2$  of a random constant  $\varepsilon_1$  and a random (Markov) process  $\varepsilon_2$  with an exponential autocorrelation:

$$\begin{aligned}\dot{\varepsilon}_1 &= 0 \\ \dot{\varepsilon}_2 &= -\varepsilon_2/\tau_\varepsilon + \tilde{v}_\varepsilon\end{aligned}\tag{12.4.2-1}$$

The accelerometer bias  $\beta$  can be modeled as the sum  $\beta = \beta_1 + \beta_2$  of two distinct random (Markov) processes  $\beta_1$  and  $\beta_2$  with exponential autocorrelations (with different decorrelation time constants)

$$\begin{aligned}\dot{\beta}_1 &= -\beta_1/\tau_1 + \tilde{v}_1 \\ \dot{\beta}_2 &= -\beta_2/\tau_2 + \tilde{v}_2\end{aligned}\quad (12.4.2-2)$$

In applications where simpler models are desirable, the above models for gyro and accelerometer errors may be simplified as two scalar Markov processes, respectively; that is,  $\varepsilon \approx \varepsilon_2$  and  $\beta \approx \beta_2$ .

The error in measuring the position  $x$  (and velocity  $\dot{x}$ ) can often be modeled as the sum of a white noise process  $w_p$  (and  $w_v$ ) and a random (Markov) process  $\Delta p$  (and  $\Delta v$ ) with an exponential autocorrelation, that is,

$$\begin{bmatrix} z_p \\ z_v \end{bmatrix} = \begin{bmatrix} x + \Delta p + w_p \\ \dot{x} + \Delta v + w_v \end{bmatrix} \quad (12.4.2-3)$$

where

$$\begin{aligned}\dot{\Delta p} &= -\Delta p/\tau_p + \tilde{v}_{\Delta p} \\ \dot{\Delta v} &= -\Delta v/\tau_v + \tilde{v}_{\Delta v}\end{aligned}\quad (12.4.2-4)$$

### 12.4.3 Single-Axis Models

Augmenting the random terms for accelerometer bias, gyro drift rate, and position and velocity measurement errors as state components, the *state-space model* for the **error state of an inertial navigation system** in a *single (horizontal) axis* becomes

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\tilde{v}(t) \quad (12.4.3-1)$$

$$z(t_k) = H\mathbf{x}(t_k) + w(t_k) \quad (12.4.3-2)$$

where the state is  $\mathbf{x} = [x, \dot{x}, \phi, \varepsilon_1, \varepsilon_2, \beta_1, \beta_2, \Delta p, \Delta v]'$ , the process noise  $\tilde{v} = [\tilde{v}_\varepsilon, \tilde{v}_1, \tilde{v}_2, \tilde{v}_{\Delta p}, \tilde{v}_{\Delta v}]'$  is a vector-valued white noise process with uncorrelated components, the measurement noise  $w = [w_p, w_v]'$  is white, and

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -g & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1/R_e & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1/\tau_\varepsilon & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1/\tau_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1/\tau_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1/\tau_p & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1/\tau_v \end{bmatrix} \quad (12.4.3-3)$$

$$B = \left[ 0_{5 \times 4} : I_{5 \times 5} \right]' \quad (12.4.3-4)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (12.4.3-5)$$

The power spectral density<sup>1</sup> of the (continuous time) process noise  $\tilde{v}$  in (12.4.3-1) is

$$\tilde{Q} = \text{diag}[0_{4 \times 4}, \text{diag}(2\sigma_\varepsilon^2/\tau_\varepsilon, 2\sigma_1^2/\tau_1, 2\sigma_2^2/\tau_2, 2\sigma_p^2/\tau_p, 2\sigma_v^2/\tau_v)] \quad (12.4.3-6)$$

while the covariance of the (discrete-time) measurement noise in (12.4.3-2) is

$$R = \text{diag}[\sigma_x^2, \sigma_{\dot{x}}^2] \quad (12.4.3-7)$$

where  $\sigma_x^2, \sigma_{\dot{x}}^2$  are the variances of  $w_p, w_v$ , respectively. In (12.4.3-6) use has been made of the fact that the power spectral density of the white noise that generates an exponentially autocorrelated noise process with instantaneous variance  $\sigma^2$  and decorrelation time constant  $\tau$  is  $2\sigma^2/\tau$  [see (8.2.2-5)].

Note that in this particular case — in the presence of the measurement noises  $w_p$  and  $w_v$  — the augmentation of the state to include the colored measurement noise terms  $\Delta p$  and  $\Delta v$  will not lead to the undesirable case of perfect (noiseless) measurements, discussed in Chapter 8, because here there is also a white noise in addition to this colored noise.

The corresponding discrete-time version of the above continuous-time dynamic model is obtained by discretization as

$$x(k+1) = Fx(k) + v(k) \quad (12.4.3-8)$$

However, the exact expression of  $F$  is not simple. An approximation that can be used for small  $T$  is

$$F = e^{AT} \approx I + AT \quad (12.4.3-9)$$

where  $T$  is the sampling interval. The covariance of the noise  $v(k)$  is given in problem 12-1.

An alternative and approximate way is to discretize the equations for  $x, \dot{x}, \phi$  and those for  $\varepsilon_1, \varepsilon_2, \beta_1, \beta_2, \Delta p, \Delta v$  separately; that is, ignore the coupling between the two groups of state components. Then

$$F \approx \text{diag}(F_{x,\dot{x},\phi}, F_{\varepsilon,\beta,\Delta p,\Delta v}) \quad (12.4.3-10)$$

where

$$F_{x,\dot{x},\phi} = e^{A_{x,\dot{x},\phi}T} = \begin{bmatrix} 1 & \frac{1}{\omega_s} \sin(\omega_s T) & R_e [\cos(\omega_s T) - 1] \\ 0 & \cos(\omega_s T) & -\frac{g}{\omega_s} \sin(\omega_s T) \\ 0 & \frac{1}{R_e \omega_s} \sin(\omega_s T) & \cos(\omega_s T) \end{bmatrix} \quad (12.4.3-11)$$

$$F_{\varepsilon,\beta,\Delta p,\Delta v} = e^{A_{\varepsilon,\beta,\Delta p,\Delta v}T} = \text{diag}(1, e^{-T/\tau_\varepsilon}, e^{-T/\tau_1}, e^{-T/\tau_2}, e^{-T/\tau_p}, e^{-T/\tau_v}) \quad (12.4.3-12)$$

and  $\omega_s = 1.24 \times 10^{-3}$  rad/s is the Schuler frequency.

---

<sup>1</sup>Often incorrectly called covariance.

#### 12.4.4 Three-Axis Models

The three-axis model is not simply a stack of three single-axis models because of the interaction among the three sensor pairs. Specifically,

$$\begin{aligned}\dot{x} &= \beta_x - g(-\phi_y) \\ -\dot{\phi}_y &= \dot{x}/R_e + \omega_x \phi_z - \varepsilon_y \\ \dot{y} &= \beta_y - g\phi_x \\ \dot{\phi}_x &= \dot{y}/R_e + \omega_y \phi_z + \varepsilon_x \\ \dot{z} &= \beta_z \\ \dot{\phi}_z &= \varepsilon_z\end{aligned}\tag{12.4.4-1}$$

where  $\omega$  is the platform angular velocity, and  $\dot{\phi}$  the corresponding error (i.e., the platform tilt rate error). Following a procedure similar to the one for the development of the single-axis model, one can develop a three-axis model for the inertial navigation system.

Under the simplifying assumption that the constant components of the accelerometer bias  $\beta$  and gyro drift rate  $\varepsilon$  have been compensated via calibration and  $[\Delta p, \Delta v]' \equiv 0$ , one has thus the following *three-axis error-state model* of an INS, modeling the remaining components of  $\beta$  and  $\varepsilon$  as white noise:

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\tilde{v}(t) \quad (12.4.4-2)$$

where the state is

$$\begin{aligned}\mathbf{x} &= [x, \dot{x}, -\phi_y, y, \dot{y}, \phi_x, z, \dot{z}, \phi_z]' \\ (x_1, x_4, x_7) &= (x, y, z) = \text{north, east, and vertically down position errors} \\ (x_2, x_5, x_8) &= (\dot{x}, \dot{y}, \dot{z}) = \text{north, east, and vertically down velocity errors} \\ (x_3, x_6, x_9) &= (-\phi_y, \phi_x, \phi_z) = \text{platform tilt angle errors about } (-y), x, z \text{ axes}\end{aligned}$$

and

$$B\tilde{v} = \begin{bmatrix} 0 \\ \beta_x \\ -\epsilon_y \\ 0 \\ \beta_y \\ \epsilon_x \\ 0 \\ \beta_z \\ \epsilon_z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_x \\ \beta_y \\ \beta_z \\ \epsilon_x \\ \epsilon_y \\ \epsilon_z \end{bmatrix} \quad (12.4.4-4)$$

$$\tilde{Q} = \text{diag}(q_x^\beta, q_y^\beta, q_z^\beta, q_x^\epsilon, q_y^\epsilon, q_z^\epsilon) \quad (12.4.4-5)$$

For a gimbaled INS,  $\beta_x, \beta_y, \beta_z, \epsilon_x, \epsilon_y, \epsilon_z$  are the actual accelerometer biases and gyro drift rates; for a strapdown INS, the actual accelerometer biases and gyro drift rates need to be converted from the body frame to the navigation coordinate system first (see next subsection). In the above, the accelerometer biases and gyro drift rates are modeled as white noise with power spectral densities  $(q_x^\beta, q_y^\beta, q_z^\beta)$  and  $(q_x^\epsilon, q_y^\epsilon, q_z^\epsilon)$ , respectively, rather than the more accurate Markov processes, as discussed in Subsections 12.4.2–12.4.3. Their values depend on the accuracy of the accelerometers and gyros used in the INS. Table 12.4.4-1 gives these power spectral density values as well as the prevailing biases. In practice, the values used in estimation filters are usually obtained by tuning.

**Table 12.4.4-1:** Biases and power spectra of INS sensors.

Sensor Parameter	Unit	INS Quality		
		High	Medium	Low
Accelerometer bias	$\mu\text{g}$	$\leq 50$	$200 - 500$	$\geq 1000$
Gyro bias	deg/hour	$\leq 0.01$	$0.1 - 1.0$	$\geq 10$
Accelerometer white noise	$(\mu\text{g})^2/\text{Hz}$	$\leq 100$	$\sim 2500$	$> 2500$
Gyro white noise	$(\text{deg}/\text{s})^2/\text{Hz}$	$\leq 10^{-9}$	$\sim 10^{-6}$	$> 10^{-6}$

The above continuous-time system is discretized to get its discrete-time counterpart. The same approach as described in the previous subsection can be used. Then, a filter can be used to obtain the state estimates of the system. The position, velocity, and attitude estimates of the vehicle at any time are then obtained by adding the filter estimates of the errors to the position, velocity, and attitude resulting from integrating the accelerometer and gyro measurements.

## 12.4.5 Coordinate Transformation

In a gimbaled system, the accelerometers are maintained, through the stabilized platform, parallel to the navigation (NAV) system axes, also known as the

geographic coordinate system, with  $x, y, z$  axes pointing to north, east, and down (toward earth center),<sup>2</sup> respectively. In a strapdown system, however, the accelerometers are either parallel or have a fixed relationship to the vehicle body axes with  $x, y, z$  axes as longitudinal (heading), lateral (right wing), and vertically down axes,<sup>3</sup> respectively, and thus their outputs have to be transformed into the navigation coordinate system in order to calculate position and velocity with respect to the earth.

The transformation from the body coordinates to the navigation coordinates is given by

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{NAV}} = C_b^n \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{Body}} \quad (12.4.5-1)$$

with

$$\begin{aligned} C_b^n &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \\ &= \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \end{aligned} \quad (12.4.5-2)$$

where

$$\psi = \text{heading (angle between the two } x \text{ axes)} = \tan^{-1}(c_{21}/c_{11})$$

$$\theta = \text{pitch (angle between the two } z \text{ axes)} = -\sin^{-1}(c_{31})$$

$$\phi = \text{roll (angle between the two } y \text{ axes)} = \tan^{-1}(c_{32}/c_{33})$$

Transformation is also needed in integrated navigation where data from different navigation systems are in general in different coordinate systems.

## 12.5 THE GLOBAL POSITIONING SYSTEM (GPS)

**GPS** is the abbreviation for *Navigation Satellite Timing and Ranging/Global Positioning System — Navstar/GPS*. It is a satellite-based navigation system with worldwide coverage that allows a user access to accurate positioning and timing information.

<sup>2</sup>Two other choices of the axes of the navigation reference coordinate systems are east-north-up and north-west-up, respectively.

<sup>3</sup>Another common choice is lateral, longitudinal, and vertically up, respectively.

GPS has been described as “the most significant development for safe and efficient navigation and surveillance of air and spacecraft since the introduction of radio navigation 50 years ago.” It has had a widespread impact, which continues to grow, due to its high accuracy, global availability, and low cost. The GPS program was formally initiated in 1973 and in 1995 the system was declared operational.

### 12.5.1 The GPS Segments

The GPS system consists of three segments (subsystems):

- Space (satellite system): Broadcast position and time signals as well as other messages.
- User (GPS receivers): Receive signal to determine position, time, and so on.
- Ground Control:
  - Update information disseminated by each satellite, monitor satellite health, control satellites, and so on.
  - Stations: Master station at Colorado Springs, several uploading stations and monitor stations

These are illustrated in Fig. 12.5.1-1.

### 12.5.2 GPS Satellite Constellation

The current constellation consists of 24 satellites in six 12-hour orbit planes (4 in each) inclined at 55 degrees to equatorial plane at a nominal altitude of 20,200 km. With this constellation, a user anywhere on the globe at any time of the day will receive signals from at least four, at most eleven, and most often six to eight satellites.

This is illustrated in Fig. 12.5.2-1.

## 12.6 GPS POSITIONING

### 12.6.1 The GPS Principle

The basic principle of GPS positioning is to obtain the distance between the user and each visible GPS satellite by measuring the time a signal travels from the satellite to the user. The user’s position (expressed in three unknown coordinates) can thus be estimated from four such distances, accounting for an additional unknown — the receiver’s clock offset  $\Delta t$  from the GPS system time. This is illustrated in Fig. 12.6.1-1 for the pseudorange positioning, where the signal traveling time is determined by the time shift required for a match between the received code from the satellite and the receiver replica (using the

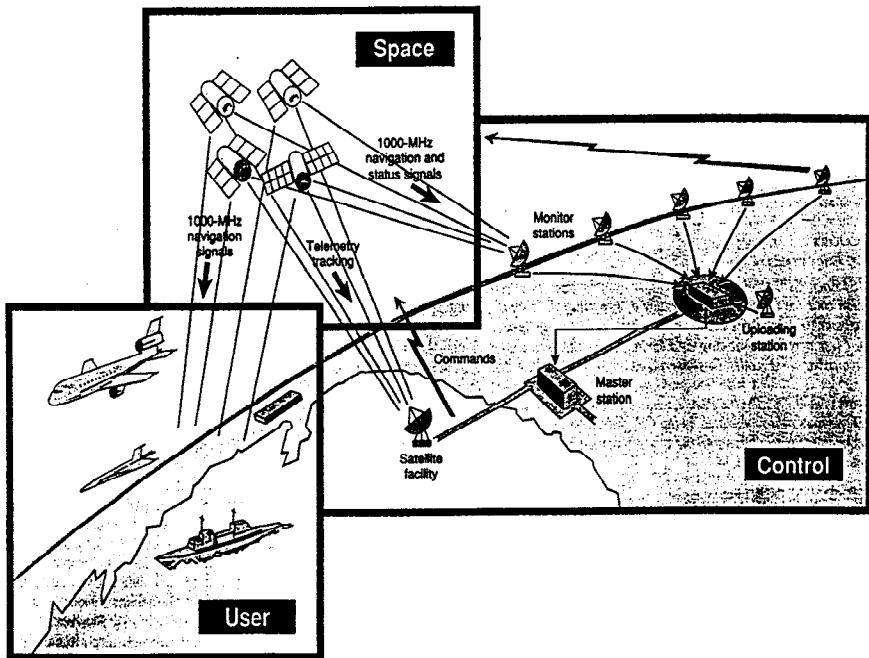


Figure 12.5.1-1: The GPS Segments (©IEEE, by permission).

start time information). Each satellite also broadcasts its clock offset ( $\tau_i$  in the figure) relative to the GPS system time so that only four unknowns need to be determined.

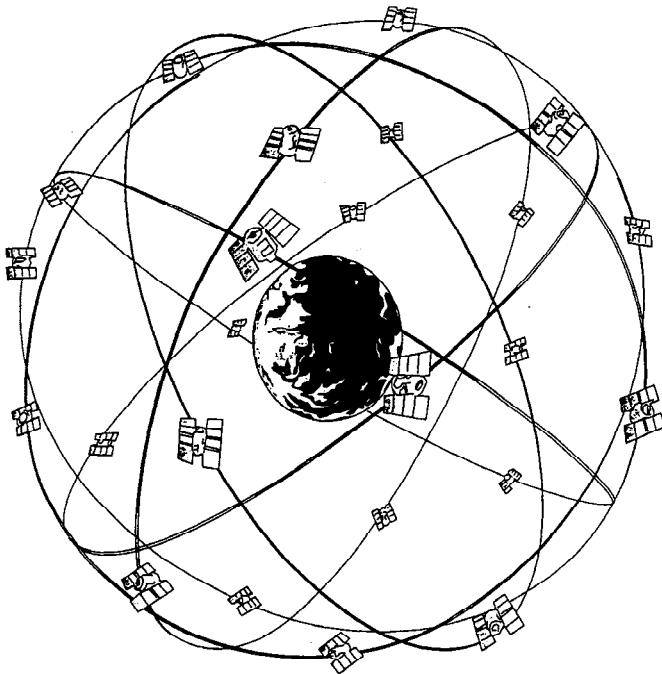
It is thus clear that at least four satellites have to be in sight. If more satellites are visible, the positioning errors due to the measurement errors can be reduced.

## 12.6.2 The GPS Signals

The signal broadcast by each satellite consists of three elements: a carrier, a unique spread-spectrum code of pseudorandom numbers, and a binary data message. It includes, among others, the following information:

- Satellite's current position
- The time the signal is sent
- Satellite's health information

GPS provides two types of service:



**Figure 12.5.2-1:** The GPS Satellite Constellation (©IEEE, by permission).

- **Standard Positioning Service (SPS):** coarse acquisition code (*C/A code*) for civilian use.
- **Precision Positioning Service (PPS):** current precision code (*P code* or *Y code*) for military use.

The unencrypted P code and the classified W code are used to generate the encrypted Y code for anti-spoofing. New military codes, referred to collectively as M code, are scheduled to be in service in a few years.

Each satellite broadcasts direct-sequence spread-spectrum signals on two carriers:

- The signal on the major carrier L1 with frequency  $f_1 = 1575.42$  MHz is

$$s_1(t) = A_x D(t) X(t) \sin(2\pi f_1 t + \theta_1) + A_p D(t) P(t) \cos(2\pi f_1 t + \theta_1) \quad (12.6.2-1)$$

where

$$X(t) = \pm 1 \quad (\text{C/A code with frequency } 1.023 \text{ MHz})$$

$$P(t) = \pm 1 \quad (\text{P code with frequency } 10.23 \text{ MHz})$$

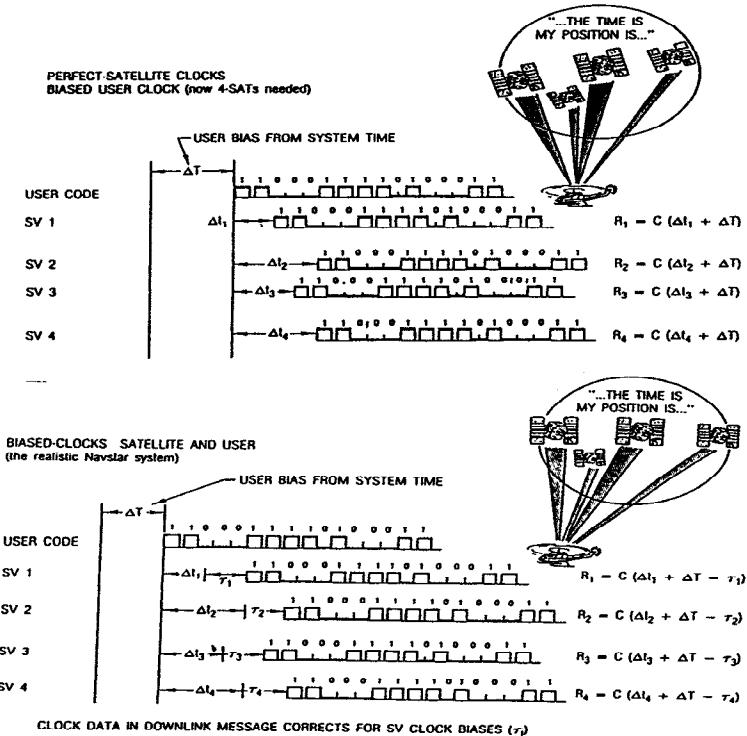


Figure 12.6.1-1: Basic principle of GPS positioning (©IEEE, by permission).

and  $D(t)$  is the navigation data, which predicts the satellite's location and clock offset relative to the GPS system time.

- The signal on the secondary carrier L2 with frequency  $f_2 = 1227.60$  MHz is

$$s_2(t) = BD(t)P(t) \cos(2\pi f_2 t + \theta_2) \quad (12.6.2-2)$$

which allows a P code user to estimate signal delay due to ionospheric and tropospheric refractions, multipath effects, etc.

Two more civil signals L2 and L5 will also be broadcast in a few years at the 1227.6 MHz and 1176.45 MHz frequencies, respectively.

The C/A and P codes have good auto- and cross-correlation properties. The autocorrelation functions have a sharp peak for zero shift so as to facilitate the match of the received and the receiver replicated codes. The various codes are nearly uncorrelated to avoid interference among signals. The spread-spectrum nature of the signals makes it possible to have a large gain in signal-to-noise ratio needed for the processing of the extremely weak GPS signals.

### 12.6.3 The GPS Observables

The basic equation for GPS positioning is the following *pseudorange equation*, for  $n$  visible satellites,

$$\rho_i = r_i + c\Delta t + w_i = \psi_i + w_i, \quad i = 1, 2, \dots, n \quad (12.6.3-1)$$

where

$\rho_i$  = pseudorange between the user and satellite  $i$

$r_i = \sqrt{(X_i - x)^2 + (Y_i - y)^2 + (Z_i - z)^2}$  = true range to satellite  $i$

$\psi_i = r_i + c\Delta t$  = noiseless pseudorange to satellite  $i$

$(X_i, Y_i, Z_i)$  = coordinates of satellite  $i$

$(x, y, z)$  = coordinates of user

$\Delta t$  = timing error (i.e., receiver clock offset from GPS time)

$c$  = speed of light ( $3 \times 10^8$  m/s)

$w_i$  = measurement error for satellite  $i$

The pseudorange measurement error  $w_i$  includes

- Errors due to Selective Availability (SA)
- Errors due to ionospheric and tropospheric refraction
- Errors due to multipath effect
- Satellite position errors
- Errors due to receiver noise, etc.

The policy of the so-called **Selective Availability (SA)** intentionally introduces two types of errors into the C/A code for signal degradation so as to prevent such unauthorized use as precision guidance of weapons: erroneous or imprecise values of the satellite ephemeris parameters, which are often referred to as “*orbit SA*,” and an erroneous drift of the GPS satellite time, known as “*dither SA*.” They affect primarily the pseudorange and delta range measurements, respectively. The SA policy has been discontinued since May 1, 2000. The prevention of the hostile use of GPS is accomplished through other measures, such as selective and local denial of GPS signals.

The pseudorange measurements are based on determining the time shift (i.e., phase difference) of the C/A and/or P codes. Measurements can also be made based on the phase (and frequency) shift between the receiver-replicated and the actually received carriers, leading to the so-called delta (pseudo)range, integrated Doppler and/or carrier phase measurements.

The GPS Doppler measurements measure the Doppler shift in carrier frequency due to the relative motion of a satellite and the user:  $\dot{r} \approx \lambda\Delta f$ , where  $\dot{r}$  is the rate of change in the (true) range (i.e., the projection of the relative velocity along the line-of-sight direction),  $\lambda$  is the known wavelength of the carrier, and  $\Delta f$  is the Doppler shift (i.e., the difference in the frequency between the

receiver replicated and the received carriers). Thus, a measurement of the range rate may be made by counting the number of Doppler cycles over a short period (e.g., 0.1 s). This is known as the *delta range* measurement. Alternatively, if the Doppler count is kept running, then a so-called *integrated Doppler, accumulated delta range*, or *continuous carrier phase measurement* is formed. Since frequency is the variation rate of phase, both of these measurements actually amount to measuring the carrier phase.

The noisy *carrier phase measurement* of each satellite is

$$\phi = \psi/\lambda + n_\phi + w_\phi \quad (12.6.3-2)$$

where

$n_\phi$  = integer ambiguity

$w_\phi$  = carrier phase measurement error

The integer ambiguity  $n_\phi$  appears because it is impossible to distinguish two carriers that differ only in phase with an integer cycle and thus only fractional-cycle phase difference is measurable. Its determination is the crux to the utilization of the carrier phase measurements. The measurement error  $w_\phi$  is made up of time-correlated errors and white measurement noise. It includes errors caused by atmospheric refraction, satellite ephemeris error, and the receiver measurement noise.

Note that the pseudorange and carrier-phase measurements in the absence of integer ambiguity obey two equations of basically the same form.

Since the carrier frequency is much higher than that of the codes, the carrier-phase measurements would be much more accurate than the pseudorange measurements if the integer ambiguity could be completely removed, which is unfortunately impossible. Also, continuous tracking of the carrier is required in the carrier-phase measurements. Interruption may occur in reality due to receiver failure or external disturbance, resulting in what is known as a *cycle slip*. Note, however, that this ambiguity does not affect the delta-range (and thus range rate) measurements.

More detailed models for pseudorange and delta-range measurements can be found in the literature; see, for example, [Braasch90, Dougherty95, Rankin94].

## 12.7 THE ACCURACY OF GPS POSITIONING

### 12.7.1 Dilution of Precision

**Dilution of Precision (DOP)** is the phenomenon that occurs when an object is localized based on distance measurements along *lines of sight (LOS)* from the object to reference points, when these LOS have small angles between them (see problem 10-10). In such a case, due to the ill-conditioning of the

estimation problem, the uncertainty in a direction perpendicular to the LOS is significantly larger than along the LOS.

The current GPS satellite constellation guarantees that four to eleven satellites are visible for a user anywhere on the globe. In general, there are more equations than unknowns. Thus the least squares solution of the equations are found below.

Linearizing the pseudorange equation yields (details given later)

$$\left. \begin{array}{l} z_1 = H_1 x + w_1 \\ \vdots \\ z_n = H_n x + w_n \end{array} \right\} \Rightarrow z = Hx + w \quad (12.7.1-1)$$

If the measurement error  $w$  has covariance  $P_w$ , then the LS estimator is

$$\hat{x} = [H' P_w^{-1} H]^{-1} H' P_w^{-1} z \quad (12.7.1-2)$$

If the measurement errors are uncorrelated and have approximately equal variance  $\sigma^2$ , then  $P_w = I\sigma^2$  and

$$\begin{aligned} \hat{x} &= [H' H]^{-1} H' z \\ P_x &= [H' H]^{-1} \sigma^2 \end{aligned} \quad (12.7.1-3)$$

where  $H$  depends only on the relative geometry of the satellites and the user.

The **Geometric DOP (GDOP) matrix** is

$$[H' H]^{-1} \triangleq \begin{bmatrix} E_{\text{DOP}}^2 & & & \\ & N_{\text{DOP}}^2 & & \\ & & V_{\text{DOP}}^2 & \\ & & & T_{\text{DOP}}^2 \end{bmatrix} \quad (12.7.1-4)$$

where, for simplicity, the off-diagonal terms are omitted, and

$$E_{\text{DOP}} = \text{east DOP}$$

$$N_{\text{DOP}} = \text{north DOP}$$

$$V_{\text{DOP}} = \text{vertical DOP}$$

$$T_{\text{DOP}} = \text{time DOP}$$

The scalar position DOP value is given by

$$\text{Position DOP} = \sqrt{\text{tr}[(H_3' H_3)^{-1}]} \quad (12.7.1-5)$$

where  $H_3$  consists of the position part<sup>4</sup> of  $H$ . The position DOP is an instantaneous measure of the observability condition and a scaling factor that relates the actual positioning accuracy and the measurement error in one direction.

---

<sup>4</sup>Note that these terms are dimensionless.

## 12.7.2 GPS Positioning Accuracy

The current nominal  $1\sigma$  magnitudes and decorrelation time constants of the major components contributing to the GPS measurement error are given in Table 12.7.2-1.<sup>5</sup> The time constants are the parameters corresponding to the correlation time of an autocorrelated (colored) random process.

Table 12.7.2-1: GPS error budget for a stationary receiver.

Error Component	Standard Deviation		Time Constant
	P Code	C/A Code	
Satellite position	2.3 m	2.3 m	1800 s
Ionospheric refraction	2 m	7 m	1800 s
Tropospheric refraction	2 m	2 m	3600 s
Multipath	1 m	2 m	600 s
Selective Availability (SA)	0	30 m	180 s

The most commonly used measures of accuracy for vertical, horizontal, and 3-D positioning are the  $2\sigma$  errors:

$$2\sigma V_{DOP} = \text{two times } \sigma \text{ times vertical DOP} \quad (12.7.2-1)$$

$$2\sigma H_{DOP} = 2\sigma \sqrt{E_{DOP}^2 + N_{DOP}^2} \quad (12.7.2-2)$$

$$2\sigma P_{DOP} = 2\sigma \sqrt{E_{DOP}^2 + N_{DOP}^2 + V_{DOP}^2} \quad (12.7.2-3)$$

where  $H_{DOP}$  = horizontal DOP,  $P_{DOP}$  = positional DOP. However, the common term “2drms” stands for “twice distance rms,” rather than “two-dimensional rms,” although the “distance” is measured in a 2-D space, the horizontal plane. Note that 2drms is not equivalent to a 95% error for a 2-D distribution. It actually corresponds to a 98% error for a circular Gaussian distribution.<sup>6</sup>

The GPS satellite constellation was designed to minimize GDOP. The upper limit for  $P_{DOP}$  was 10. The worldwide daylong median values are  $P_{DOP} = 2.0$  and  $H_{DOP} = 1.2$ . For a normal situation,  $P_{DOP} \leq 4$ .

The actual positioning accuracies by code-phase measurements are approximately as follows:

- The accuracies of the PPS (P code) are: 17.8 m (2drms, i.e., 98%) horizontal, 27.7 m ( $2\sigma$ , i.e., 95%) vertical, and 90 ns maximum timing error.

<sup>5</sup>Differential GPS (DGPS), or the Wide Area Augmentation System (WAAS), which use local stations that transmit corrections (valid within their local regions) for the slowly time-varying errors, can improve the overall accuracy by an order of magnitude or more.

<sup>6</sup>For a circular Gaussian with s.d.  $\sigma$  in each direction, drms =  $\sqrt{2}\sigma$ , and  $(2\text{drms})^2 = 8\sigma^2$ . Using the 2 degrees of freedom chi-square distribution tables, the threshold 8 corresponds to 98% probability (2% tail).

- The specified accuracies of the SPS (C/A code with SA) are 100 m (2drms) horizontal, 156 m ( $2\sigma$ ) vertical,<sup>7</sup> and 175 ns maximum timing error. If the median values are used, the  $2\sigma$  accuracies are

$$\begin{aligned} \text{3-D accuracy: } & 2\sigma P_{\text{DOP}} = 2 \times 30 \times 2.0 = 120 \text{ m} \\ \text{horizontal accuracy: } & 2\sigma H_{\text{DOP}} = 2 \times 30 \times 1.2 = 72 \text{ m} \end{aligned}$$

These nominal values are greatly reduced in the absence of SA and further reduced when the new civil signals L2 and L5 are broadcast.

Some manufacturers specify the positioning errors of their GPS receivers in terms of the so-called circular error probable (CEP) [or spherical error probable (SEP)], which correspond to 50% probability mass. The  $1\sigma$  error corresponds, for a one-dimensional Gaussian, to 68% probability mass. These and several other commonly used accuracy measures are listed in Table 12.7.2-2. The notation  $[\chi_n^2]^{-1}(\cdot)$  denotes the inverse function of (1.5.4-3), which yields the probability masses corresponding to the various accuracy definitions.

**Table 12.7.2-2:** Radius and probability mass corresponding different accuracy units for a circular Gaussian with unit s.d. in each coordinate.

	Radius	Probability Mass
Vrms (vertical rms)	1	$[\chi_1^2]^{-1}(1) = 0.68$
CEP	$\sqrt{1.4} = 1.18$	$[\chi_2^2]^{-1}(1.4) = 0.50$
Hrms (horizontal rms)	$\sqrt{2} = 1.41$	$[\chi_2^2]^{-1}(2) = 0.63$
R95 (horizontal 95%)	$\sqrt{6} = 2.45$	$[\chi_2^2]^{-1}(6) = 0.95$
2drms	$2\sqrt{2} = 2.83$	$[\chi_2^2]^{-1}(8) = 0.98$
SEP	$\sqrt{2.4} = 1.55$	$[\chi_3^2]^{-1}(2.4) = 0.50$
3-D rms	$\sqrt{3} = 1.73$	$[\chi_3^2]^{-1}(3) = 0.61$

The positioning accuracy by carrier-phase measurements and its integration with the code-phase measurements is at the sub-meter (could be centimeter) level.

There are no published specifications for velocity estimates. They are obtained internally by the GPS receiver by filtering the measurements, as shown later.

<sup>7</sup>Since this is 1-D, the  $2\sigma$  region encompasses 95% probability.

## 12.8 STATE-SPACE MODELS FOR GPS

### 12.8.1 Models for Receiver Clock State

As can be seen from (12.6.3-1), the unknown timing error  $\Delta t$  due to user receiver clock offset from the GPS system time comes into play in the GPS navigation solution. Quite often, two state components are used to model this offset: clock bias  $b = \Delta t$  and clock drift  $d$ , which represent the phase and frequency errors in the receiver clock, respectively. They are usually modeled as

$$\dot{b}(t) = d(t) + \tilde{v}_b(t) \quad (12.8.1-1)$$

$$\dot{d}(t) = \tilde{v}_d(t) \quad (12.8.1-2)$$

where  $\tilde{v}_b(t)$  and  $\tilde{v}_d(t)$  are mutually independent, zero-mean white noise with power spectra  $S_b$  and  $S_d$ , respectively. Its discretized version is

$$x_c(k+1) = F_2 x_c(k) + v_c(k) \quad \text{with} \quad F_2 = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad (12.8.1-3)$$

where  $x_c = [b \ d]'$  and  $v_c(k)$  has zero mean and the covariance matrix (see problem 12-3)

$$Q_c = S_b T \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + S_d V_2, \quad V_2 = \begin{bmatrix} \frac{1}{2}T^3 & \frac{1}{2}T^2 \\ \frac{3}{2}T^2 & T \end{bmatrix} \quad (12.8.1-4)$$

$T$  is the sampling interval.

The spectral densities  $S_b$  and  $S_d$  are, in general, filter design parameters. For many GPS clocks, however, they can be set approximately as

$$S_b = (0.5 \sim 2)\gamma \quad S_d = (2 \sim 8)\pi^2\alpha \quad (12.8.1-5)$$

where  $\alpha$  and  $\gamma$  are the Allan variance parameters that are often used to describe clock offset. Typical values of these parameters are given in Table 12.8.1-1.

Table 12.8.1-1: Typical Allan variance parameters for various timing standards.

Clock Type	$\alpha$	$\gamma$
Crystal	$2 \times 10^{-20}$	$2 \times 10^{-19}$
Ovenized crystal	$4 \times 10^{-23}$	$8 \times 10^{-20}$
Rubidium	$4 \times 10^{-29}$	$2 \times 10^{-20}$

### 12.8.2 Dynamic Models

A proper dynamic model depends on the actual motion of the vehicle. What is special for the GPS application is that the state should be augmented to include components for the modeling of errors due to clock offset and other autocorrelated processes for the modeling of measurement errors.

For example, in the absence of SA and other autocorrelated terms in the measurement error that can be modeled as Markov processes, an observer with a uniform motion (i.e., with nearly constant altitude and nearly constant horizontal velocity) can be described by

$$\mathbf{x}(k+1) = F\mathbf{x}(k) + v(k) \quad (12.8.2-1)$$

with the state  $\mathbf{x} = [x \ \dot{x} \ y \ \dot{y} \ z \ x'_c]'$ , where  $v(k)$  is the equivalent discrete-time noise of the continuous-time process noise  $\tilde{v}(t) = [0 \ \tilde{v}_x \ 0 \ \tilde{v}_y \ \tilde{v}_z \ \tilde{v}'_c]'$ ;  $(x, y, z)$  is the position; and  $x_c$  and  $\tilde{v}_c$  are the user clock state and the associated noise defined in (12.8.1-3). All noises are assumed white. The state transition matrix is

$$F = \text{diag}[F_2 \ F_2 \ 1 \ F_2] \quad (12.8.2-2)$$

The corresponding covariance of the process noise can be shown to be

$$Q = \text{diag}[S_{\dot{x}}V_2 \ S_{\dot{y}}V_2 \ S_zT \ Q_c] \quad (12.8.2-3)$$

where  $V_2$  and  $Q_c$  are given by (12.8.1-4).

The power spectral densities  $S_{\dot{x}}$ ,  $S_{\dot{y}}$ , and  $S_z$  of  $\tilde{v}_x$ ,  $\tilde{v}_y$ ,  $\tilde{v}_z$ , respectively, are, in general, filter design parameters. They can be determined via an “educated guess” based on the expected vehicle dynamics, according to the discussions in Subsection 6.2.2.

Several other typical vehicle motion models are considered in problem 12-4.

### Remarks

Note that unlike INS, GPS does not provide vehicle attitude information directly and thus the state usually does not include attitude components.

The multiple-model estimation approach may be considered in some cases, such as for navigation of a highly dynamic vehicle without an INS, or for cases in the presence of uncertainty of satellite failures. An example is given in the next section.

### 12.8.3 Linearized Measurement Model

To use the (extended) Kalman filtering, the measurement equation (12.6.3-1) should be linearized around the prediction  $\hat{x}(k|k-1)$  as follows

$$z(k) = \psi[\mathbf{x}(k)] + w(k)$$

$$\begin{aligned} &\approx \psi[\hat{\mathbf{x}}(k|k-1)] + \frac{\partial\psi}{\partial\mathbf{x}}\Big|_{-\hat{\mathbf{x}}(k|k-1)}[\mathbf{x}(k) - \hat{\mathbf{x}}(k|k-1)] + w(k) \\ &= \hat{z}(k|k-1) + H(k)[\mathbf{x}(k) - \hat{\mathbf{x}}(k|k-1)] + w(k) \end{aligned} \quad (12.8.3-1)$$

which leads to the linear pseudo-measurement equation for the Kalman filter:

$$z(k)^* \triangleq z(k) - \psi[\hat{\mathbf{x}}(k|k-1)] + H(k)\hat{\mathbf{x}}(k|k-1) = H(k)\mathbf{x}(k) + w(k) \quad (12.8.3-2)$$

In the above, if  $n$  satellites are visible and  $\mathbf{x} = [x, \dot{x}, y, \dot{y}, z, x'_c]'$ , then

$$H = \frac{\partial\rho}{\partial\mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}} = \frac{\partial\psi}{\partial\mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}} = \begin{bmatrix} -h_1^x & 0 & -h_1^y & 0 & -h_1^z & c & 0 \\ -h_2^x & 0 & -h_2^y & 0 & -h_2^z & c & 0 \\ \vdots & & & & & & \vdots \\ -h_n^x & 0 & -h_n^y & 0 & -h_n^z & c & 0 \end{bmatrix} \quad (12.8.3-3)$$

in which  $(h_i^x, h_i^y, h_i^z)$  is the user-to-satellite- $i$  LOS vector and  $c$  is the speed of light. The elements of this vector, known as *direction cosines*, are given by

$$h_i^x = -\frac{\partial\psi_i}{\partial x} = \frac{(X_i - x)}{\sqrt{(X_i - x)^2 + (Y_i - y)^2 + (Z_i - z)^2}} \quad (12.8.3-4)$$

$$h_i^y = -\frac{\partial\psi_i}{\partial y} = \frac{(Y_i - y)}{\sqrt{(X_i - x)^2 + (Y_i - y)^2 + (Z_i - z)^2}} \quad (12.8.3-5)$$

$$h_i^z = -\frac{\partial\psi_i}{\partial z} = \frac{(Z_i - z)}{\sqrt{(X_i - x)^2 + (Y_i - y)^2 + (Z_i - z)^2}} \quad (12.8.3-6)$$

As a result, the GDOP matrix  $(H'H)^{-1}$  is actually not diagonal and the correct version of scalar position DOP is given by  $\sqrt{\text{tr}[(H_p'H_p)^{-1}]}$ , where  $H_p$  is the position part of  $H$ , given by

$$H'_p = \begin{bmatrix} -h_1^x & -h_2^x & \cdots & -h_n^x \\ -h_1^y & -h_2^y & \cdots & -h_n^y \\ -h_1^z & -h_2^z & \cdots & -h_n^z \end{bmatrix} \quad (12.8.3-7)$$

The covariance matrix of the measurement error is assumed to be diagonal:  $R = \text{cov}[w] = \text{diag}[\sigma_\rho^2 \ \cdots \ \sigma_\rho^2]$ , i.e., uncorrelated across satellites.

Linearization of the carrier-phase measurements is done similarly.

## 12.8.4 A Model for Exponentially Autocorrelated Noise

The measurement noise  $w$  accounts for the effect of the various error components (see Table 12.7.2-1). Assuming it is a stationary first-order Markov process, each component can be modeled as the state  $w_e$  of a scalar LTI prewhitening system (shaping filter — see Subsection 1.4.21) driven by zero-mean white noise  $w_e^w$  as follows:

$$w_e(k+1) = \lambda w_e(k) + w_e^w(k) \quad (12.8.4-1)$$

The original dynamic equations should be augmented by this equation for  $w_e$  in the Kalman filter. The variance  $q_e^w = E[w_e^w(k)^2]$  of the corresponding process noise  $w_e^w$  is found next.

Assuming that the dominant error component (without SA) has an RMS value of 3 m and a time constant  $T_c = 600$  s,<sup>8</sup> one has, with  $T = 1$ ,

$$\lambda = e^{-T/T_c} = e^{-1/600} = 0.9983 \quad (12.8.4-2)$$

Hence

$$E[w_e(k+1)^2] = E\{[\lambda w_e(k) + w_e^w(k)]^2\} = \lambda^2 E[w_e(k)^2] + E[w_e^w(k)^2] \quad (12.8.4-3)$$

Consequently, in view of the stationarity of  $w_e$ , i.e.,

$$E[w_e(k+1)^2] = E[w_e(k)^2] = \sigma_e^2 = (3m)^2 \quad (12.8.4-4)$$

one has

$$\begin{aligned} (\sigma_e^w)^2 &= E[w_e^w(k)^2] = (1 - \lambda^2)E[w_e(k)^2] \\ &= (1 - 0.9983^2)(3^2) = (0.0306 \text{ m})^2 \end{aligned} \quad (12.8.4-5)$$

To avoid the singularity (perfect, noiseless measurement) case associated with a direct state augmentation to account for the colored measurement, it is better assumed that the pseudorange measurement noise  $w(k)$  actually is a sum of a Markov noise sequence  $w_e(k)$ , as modeled above, and a white noise sequence  $w^w(k)$  that accounts for other unmodeled measurement errors.

The total noise (for each pseudorange measurement) is then

$$w(k) = w_e(k) + w^w(k) \quad (12.8.4-6)$$

This equation should be substituted into (12.8.3-1) to yield the corresponding linearized measurement equation.

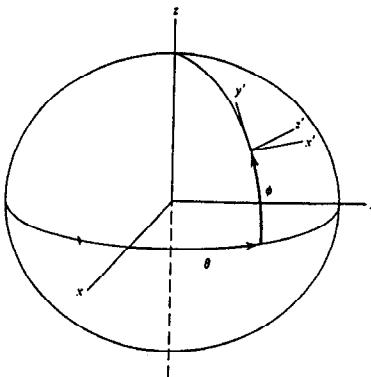
Note that the dimension of the filter increases by  $n$  (the number of visible satellites) because each pseudorange measurement noise requires a first-order shaping filter.

## Remarks

Even though the actual standard deviation of the pseudorange measurement noise is rather small (about 1m for P code and several meters for C/A code) in the absence of Selective Availability, a slightly larger value is often used to account for other unmodeled measurement errors.

---

<sup>8</sup>This models (approximately) a P code or a C/A code with local corrections (DGPS).



**Figure 12.8.5-1:** The WGS-84 ( $x, y, z$ ) and navigation ( $x', y', z'$ ) reference coordinate system.

### 12.8.5 Coordinate Transformation

The satellite position signals are in the *World Geodetic System of 1984 (WGS-84)* coordinate system, which is shown in Fig. 12.8.5-1. Usually one is more interested in the navigation reference frame and thus a transformation is needed. By concatenating a series of linear translational and rotational transformations similar to the one in Subsection 12.4.5, a linear transformation from the WGS-84 system to the navigation reference (east, north, up) coordinate system can be obtained as

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}_{\text{NAV}} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{WGS-84}} = \begin{bmatrix} -\sin \theta & \cos \theta & 0 \\ -\sin \phi \cos \theta & -\sin \phi \sin \theta & \cos \phi \\ \cos \phi \cos \theta & \cos \phi \sin \theta & \sin \phi \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{WGS-84}} + \begin{bmatrix} x_0 \sin \theta - y_0 \cos \theta \\ x_0 \sin \phi \cos \theta + y_0 \sin \phi \sin \theta - z_0 \cos \phi \\ -x_0 \cos \phi \cos \theta - y_0 \cos \phi \sin \theta - z_0 \sin \phi \end{bmatrix} \quad (12.8.5-1)$$

where  $\phi$  and  $\theta$  are the local reference latitude and longitude, respectively, and  $(x_0, y_0, z_0)$  are the coordinates of the origin of the navigation reference coordinate system in the WGS-84 system.

## 12.9 EXAMPLE: GPS NAVIGATION WITH IMM ESTIMATOR

This example demonstrates the following:

- The superiority of using IMM estimator to the Kalman filter in some dynamic positioning applications
- A simple procedure to generate the GPS pseudorange measurements for the convenience of the simulation study of GPS applications

### 12.9.1 Generation of Satellite Trajectories

To simulate GPS scenarios, a description of typical satellite orbit trajectories is needed that quantifies the satellite geometries. However, an exact description is unnecessarily too complicated for most simulation studies.

For simplicity, assume that the satellite orbit is circular and the earth is round. The position coordinates of a satellite in the WGS-84 coordinate system are then given by

$$\begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix}_{\text{WGS-84}} = \begin{bmatrix} R[\cos \omega(t) \cos \Omega(t) - \sin \omega(t) \sin \Omega(t) \cos \alpha] \\ R[\cos \omega(t) \sin \Omega(t) + \sin \omega(t) \cos \Omega(t) \cos \alpha] \\ R \sin \omega(t) \sin \alpha \end{bmatrix} \quad (12.9.1-1)$$

where  $R$  is the radius of the circular orbit;  $\alpha$  is the inclination angle between the orbital plane and the equatorial plane;  $\Omega$  is the so-called right ascension, that is, the longitude of the point on the orbit where the orbital plane intersects the equatorial plane as the orbit crosses over from the south to the north; and  $\omega$  is the angle of the satellite in the circular orbit relative to the right ascension. They are given by

$$\omega(t) = \omega_0 + \frac{360}{43082}(t - t_0)^\circ \quad (12.9.1-2)$$

$$\Omega(t) = \Omega_0 - \frac{360}{86164}(t - t_0)^\circ \quad (12.9.1-3)$$

$$\alpha = 55^\circ \quad (12.9.1-4)$$

$$R = 26,560 \text{ km} \quad (12.9.1-5)$$

The periodical variation of  $\Omega$  is due to the rotation of the earth and the period 86,164 seconds is approximately equal to 24 hours. Such coordinates are in the WGS-84 system and should be converted to the navigation reference system using (12.8.5-1). If the origin of the navigation reference coordinate system is at the “zero-zero” point ( $\phi = \theta = 0$ ) of the WGS-84 system, that is,  $(x, y, z)_{\text{NAV}} = (y, z, x - r)$ , where  $(x, y, z)$  are the coordinates in the WGS-84 and  $(x, y, z)_{\text{NAV}}$  are the coordinates (east, north, up) in the navigation systems,  $r = 6360$  km is the earth radius, then the coordinates of the satellite in the navigation coordinate system are

$$\begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix}_{\text{NAV}} = \begin{bmatrix} R[\cos \omega(t) \sin \Omega(t) + \sin \omega(t) \cos \Omega(t) \cos \alpha] \\ R \sin \omega(t) \sin \alpha \\ R[\cos \omega(t) \cos \Omega(t) - \sin \omega(t) \sin \Omega(t) \cos \alpha] - r \end{bmatrix} \quad (12.9.1-6)$$

The “initial” positions (angles  $\Omega_0$  and  $\omega_0$ ) of a group of satellites that constitute

a typical scenario of satellite geometries are given in the following table:

Satellite	$\Omega_0(^{\circ})$	$\omega_0(^{\circ})$
1	325.7	72.1
2	25.7	343.9
3	85.7	214.9
4	145.7	211.9
5	205.7	93.9
6	265.7	27.9

The position DOP of this scenario for a user located around the “zero–zero” location is approximately equal to 3.5. In order for all the satellites to be visible, one has to have  $t_0 = -3000$  s.

The above is a simplified model for a typical GPS scenario. It neglects several features that are important for some applications. More detailed descriptions are available, see e.g., [Carvalho97].

### 12.9.2 Generation of Trajectories and Pseudorange Measurements

Consider a user (a platform) equipped with a GPS receiver that has access to C/A code but not P code. At initial time  $t = 0$ , the platform was at

$$\mathbf{x}_p(0) = [x \quad \dot{x} \quad y \quad \dot{y} \quad z]' = [0 \quad 0 \quad 0 \quad 30 \text{ m/s} \quad 0]'$$

where  $x$ ,  $y$ , and  $z$  are the coordinates (in m) of the navigation reference coordinate system. The (average) motion of the platform is

- $t \in (0, 50]$ : constant velocity  $[0 \quad 30 \text{ m/s}]$  in the horizontal  $x$ - $y$  plane due east
- $t \in [50, 90]$ : a  $3^{\circ}/\text{s}$  left “coordinated turn” (a constant speed and angular rate turn, with lateral acceleration of  $0.15g$ ) of  $120^{\circ}$  in the horizontal plane
- $t \in [90, 150]$ : constant velocity
- $t \in [150, 180]$ : a  $9^{\circ}/\text{s}$  right “coordinated turn” (lateral acceleration of  $0.45g$ ) of  $120^{\circ}$  in the horizontal  $x$ - $y$  plane
- $t \in [180, 250]$ : constant velocity

The actual motion of the platform is the superposition of the motion described above and the random motion due to turbulence, which is modeled as a continuous-time zero-mean white Gaussian noise whose components represent the random and uncorrelated accelerations along  $x$  and  $y$  directions and velocity along  $z$  direction, respectively. The true covariance matrix of the equivalent discrete-time noise (not known to the filter) is assumed to be, for the motion part,

$$Q = \text{diag}[S_{\dot{x}}V_2 \quad S_{\dot{y}}V_2 \quad TS_z] \quad (12.9.2-1)$$

where  $V_2$  was given in Subsection 12.8.2,  $S_{\dot{x}} = S_{\dot{y}} = 0.01 \text{ m}^2/\text{s}^3$ , and  $S_z = 0.01 \text{ m}^2/\text{s}$ .

The GPS receiver equipped with a *crystal clock* receives synchronized signals from all visible satellites every one second, i.e., the sampling interval is  $T = 1 \text{ s}$ . The time constant for the dominant error component is assumed to be  $T_c = 600 \text{ s}$ .

The pseudorange measurement from each satellite is generated from (12.6.3-1), with the clock state  $x_c = [b \ d]'$  obtained from the following equation

$$x_c(k+1) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x_c(k) + v_c(k) \quad (12.9.2-2)$$

where  $v_c(k)$  is white noise with covariance  $Q_c$ , given in Subsection 12.8.2.

The pseudorange measurement noise  $w(k)$  is assumed to be the sum of a first-order Gauss-Markov (exponentially autocorrelated) sequence  $w_e(k)$  and a white noise  $w^w(k)$ , the latter with covariance  $R = \sigma^2 I_{n \times n}$ , where  $n$  is the number of visible satellites. The elements of  $w_e(k)$  are uncorrelated, and each of them satisfies (12.8.4-1). The covariances

$$\text{cov}[w_e(k)] = \sigma_e^2 I_{n \times n} = (3)^2 I_{n \times n} \quad (12.9.2-3)$$

$$\text{cov}[w^w(k)] = \sigma^2 I_{n \times n} = (1)^2 I_{n \times n} \quad (12.9.2-4)$$

were used to obtain the results to be presented below.

### 12.9.3 State-Space Models

Two IMM estimators, IMM-L and IMM-CT, and two Kalman filters, KF1 and KF2, are used for comparison. IMM-L estimator uses two WNA motion models with two process noise levels to cover uniform motion and maneuver, respectively. IMM-CT uses one WNA motion model and a nearly coordinated turn (CT) model. The extended Kalman filters KF1 and KF2, each based on a single WNA motion model, are used as baselines.

The state-space models of the IMM-L estimator and the baseline Kalman filters have the following form:

$$\mathbf{x}(k+1) = F\mathbf{x}(k) + v(k) \quad (12.9.3-1)$$

$$z(k) = h[k, \mathbf{x}(k)] + w^w(k) \quad (12.9.3-2)$$

where  $w^w$  is the white noise part of the pseudorange measurement error,  $\mathbf{x}$  is the augmented state

$$\mathbf{x} = [x \ \dot{x} \ y \ \dot{y} \ z \ x'_c \ w_{e_1} \ \dots \ w_{e_n}]' \quad (12.9.3-3)$$

and

$$\begin{aligned} h[k, \mathbf{x}(k)] &= r_i(k) + b(k) + w_{e_i}(k) \\ &= \sqrt{(X_i(k) - x(k))^2 + (Y_i(k) - y(k))^2 + (Z_i(k) - z(k))^2} \\ &\quad + b(k) + w_{e_i}(k) \end{aligned} \quad (12.9.3-4)$$

$(X_i, Y_i, Z_i)$  are the coordinates of the  $i$ th satellite,  $b$  is the receiver clock bias, and  $w_{e_i}$  is the autocorrelated (Markov process) part of the pseudorange measurement error. The linearized measurement system matrix is

$$\begin{aligned} H &= \frac{\partial}{\partial \mathbf{x}} h[k, \mathbf{x}(k)]|_{\mathbf{x}(k)=\hat{\mathbf{x}}(k)} \\ &= \begin{bmatrix} -h_1^x & 0 & -h_1^y & 0 & -h_1^z & c & 0 \\ -h_2^x & 0 & -h_2^y & 0 & -h_2^z & c & 0 \\ \vdots & & & & & \vdots & I_{n \times n} \\ -h_n^x & 0 & -h_n^y & 0 & -h_n^z & c & 0 \end{bmatrix}_{\mathbf{x}(k)=\hat{\mathbf{x}}(k)} \end{aligned} \quad (12.9.3-5)$$

where  $h_i^x, h_i^y, h_i^z$  were given in Subsection 12.7.1, and the measurement prediction is

$$\hat{z}(k|k-1) = h[k, \hat{\mathbf{x}}(k|k-1)] \quad (12.9.3-6)$$

The covariance of the process noise used in the filter has the form

$$Q_f = \text{diag}[qV_2 \quad qV_2 \quad TS_z \quad Q_c \quad (\sigma_e^w)^2 I_{n \times n}] \quad (12.9.3-7)$$

where  $q$  is a design parameter and  $\sigma_e^w$  is determined as in (12.8.4-5).

The IMM-L estimator and the Kalman filters assume that the platform is in a nearly uniform motion (i.e., straight and level nearly constant velocity motion) with the following augmented state system matrix:

$$F = \text{diag}[F_2 \quad F_2 \quad 1 \quad F_2 \quad \lambda I_{n \times n}] \quad (12.9.3-8)$$

where  $\lambda$  is given in (12.8.4-2). The Kalman filters use an artificially inflated covariance  $Q_f$  of the process noise to cover possible deviations from the uniform motion, such as the turns.

The state-space model of IMM-CT is similar to the above by augmenting the state vector one more component — the turn rate  $\Omega$ , that is,

$$\mathbf{x}_{CT} = [x \quad \dot{x} \quad y \quad \dot{y} \quad \Omega \quad z \quad x'_c \quad w_{e_1} \quad \cdots \quad w_{e_n}]' \quad (12.9.3-9)$$

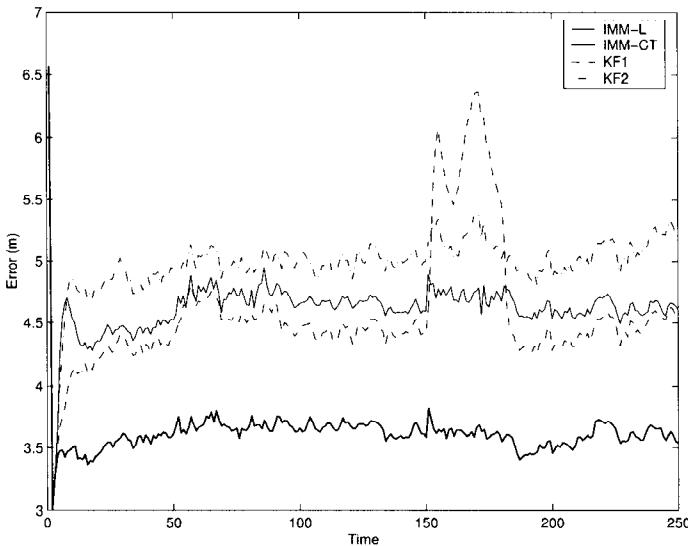
$$F = \text{diag}[F_2 \quad F_2 \quad 1 \quad 1 \quad F_2 \quad \lambda I_{n \times n}] \quad (12.9.3-10)$$

For the IMM-L estimator, the process noise power spectral densities are  $q_1 = 0.01 \text{ m}^2/\text{s}^3$  and  $q_2 = 9 \text{ m}^2/\text{s}^3$  for the quiescent and maneuvering model, respectively.<sup>9</sup> The Markov chain transition matrix between the modes is

$$\pi = \begin{bmatrix} 0.90 & 0.10 \\ 0.10 & 0.90 \end{bmatrix} \quad (12.9.3-11)$$

The performance of the IMM estimator is not very sensitive to the choice of these transition probabilities. However, this choice provides to a certain degree

<sup>9</sup>Note that for the discretized WNA model the corresponding RMS acceleration is  $\sqrt{g/T}$ . The above two values correspond to  $0.1 \text{ m/s}^2$  and  $3 \text{ m/s}^2$ , respectively. For the vertical motion  $q_z = 0.01 \text{ m}^2/\text{s}$ , which corresponds to a motion RMS uncertainty in position over  $T = 1 \text{ s}$  of  $\sqrt{q_z T} = 0.1 \text{ m}$ .



**Figure 12.9.4-1:** RMS position errors.

the trade-off between the peak estimation errors at the onset (termination) of the maneuver and the maximum reduction of the estimation errors during the uniform motion.

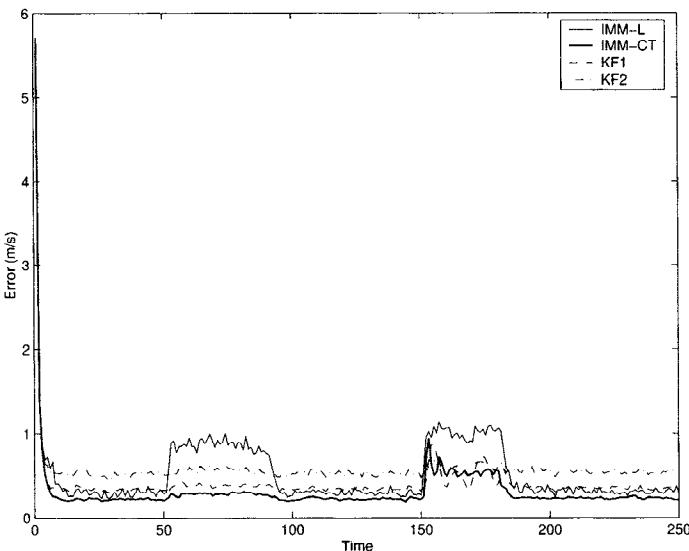
For the IMM-CT estimator, the process noise power spectral densities are  $0.01 \text{ m}^2/\text{s}^3$  for both uniform motion model and coordinate turn model, and the standard deviation of the turn rate process noise is  $3^\circ/\text{s}^2$ . The transition matrix is the same as above.

The process noise power spectral densities in the two versions of the Kalman filter KF1 and KF2 are  $0.25 \text{ m}^2/\text{s}^3$  and  $1 \text{ m}^2/\text{s}^3$ . The smaller process noise level will yield a better performance during the uniform motion at the cost of a higher estimation error during the maneuver, i.e., a trade-off is needed in practice.

## 12.9.4 Simulation Results and Discussion

The simulation results are based on 200 Monte Carlo runs. The RMS position errors for these four estimators are presented in Fig. 12.9.4-1. Note how KF1 (with low process noise) has a lower error during the quiescent period but it shoots up during maneuvers; KF2 is a compromise.

The RMS speed errors and RMS course errors are shown on Fig. 12.9.4-2 and Fig. 12.9.4-3, respectively. The IMM estimators have better performances. The IMM-CT estimator has under  $0.3 \text{ m/s}$  speed estimation errors during uniform motion and the first turn. The probabilities of the maneuvering mode in the



**Figure 12.9.4-2:** RMS speed errors.

IMM estimators are shown on Fig. 12.9.4-4. We can see very clearly the “soft switching” between two models in the IMM estimators.

Table 12.9.4-1 shows that IMM-L has almost the same average RMS position performance during the quiescent period as to the best Kalman filter (KF1), but 30% lower error during maneuvers. IMM-CT is uniformly better than all the others, especially in speed and course errors. The peak RMS errors are shown in Table 12.9.4-2.

The reason why the IMM estimator outperforms the KF is the following. For the KF, a *compromise* choice of the process noise variance has to be used in order to have acceptable accuracy for both the quiescent and maneuvering periods. For the IMM estimator, since more than one model can be used and each model is fairly accurate, the algorithm will automatically give more weight to the prevailing model, i.e., it will adapt itself to the motion.

The following observations can be made from the figures and tables.

- The IMM estimators and the Kalman filters have about the same accuracy for uniform motion.
- The IMM estimators have much better speed estimates and significantly better course estimates than the Kalman filters during the turns.
- The IMM estimators have much better position estimates than the Kalman filters during the turns except for the beginning and end of the turns at which they are close.

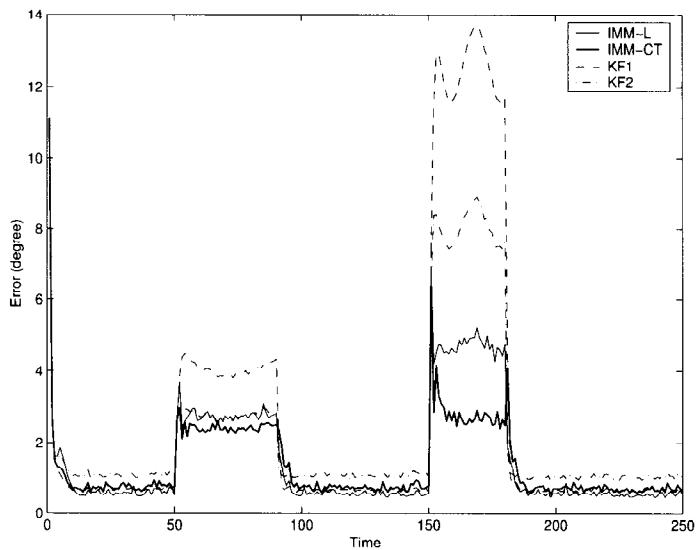


Figure 12.9.4-3: RMS course errors.

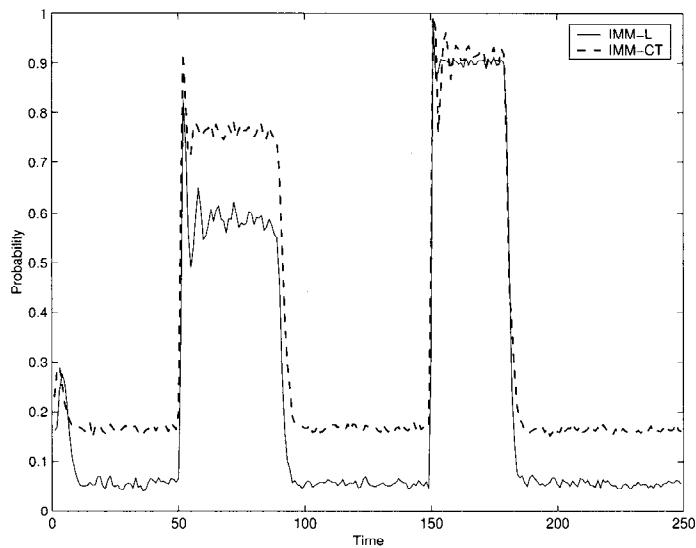


Figure 12.9.4-4: Maneuvering mode probabilities.

**Table 12.9.4-1:** Average quiescent (maneuvering) period RMS errors.

	KF1	KF2	IMM-L	IMM-CT
Position (m)	4.4 (5.1)	5.0 (5.1)	4.6 (4.7)	3.6 (3.7)
Speed (m/s)	0.3 (0.4)	0.5 (0.6)	0.3 (0.9)	0.2 (0.4)
Course (°)	0.7 (7.6)	1.1 (5.0)	0.6 (3.6)	0.8 (2.6)

**Table 12.9.4-2:** Peak RMS errors.

	KF1	KF2	IMM-L	IMM-CT
Position (m)	6.4	5.4	4.9	3.8
Speed (m/s)	0.9	0.7	1.1	0.9
Course (°)	13.7	8.9	7.0	6.4

### 12.9.5 Do We Need an IMM Estimator for GPS?

As shown recently in [Kirubarajan00] (see Section 11.8), the IMM estimator is superior to a Kalman filter (which, being based on a single model is, by necessity, a compromise filter between the quiescent and maneuvering situations) when the maneuvering index exceeds 0.5.

If GPS is aided by an INS, there is relatively little motion uncertainty error because the platform accelerations are measured directly by the INS system. However, without INS — i.e., in a (low-cost and light) GPS stand-alone system — one can have a maneuvering index that exceeds 0.5. Interestingly, *as the position measurements become more accurate, the maneuvering index increases and there is more opportunity to enhance the estimation accuracy by replacing the Kalman filter with a suitable IMM estimator* [Lin01].

## 12.10 INTEGRATED NAVIGATION

Compared with other navigation systems, the INS has two primary advantages: self-containment and high accuracy for short-term navigation. Its most serious disadvantage is that it exhibits position errors that grow unbounded with time, caused by the accumulation of gyro and accelerometer errors over time, as well as oscillatory velocity errors. Mainly for this reason, in long-term navigation applications the INS is often used in conjunction with various navigation aids, including radio navigation systems (e.g., Loran, Omega, and Tacan), satellite navigation systems (e.g., GPS, GLONASS, and Transit), and others (e.g., JTIDS and DME). These navigation aids provide navigation information that does not

deteriorate over time and thus are more advantageous for long-term navigation applications. To take full advantage of the INS and navigation aids, a good integration mechanism should be implemented.

GPS and INS are almost ideally complementary to each other in that GPS provides measurements in the position-time and velocity-frequency domains and INS provides measurements of acceleration and vehicle attitude, in addition to their complementary advantages for short- and long-term navigation. Also, GPS is gaining momentum rapidly. This section will, therefore, focus on the **integration of INS and GPS**. However, the methodologies and discussions are applicable to other **integrated navigation** applications.

If one system is treated as a primary navigation means, there are two general approaches to the integration of INS and GPS: **GPS-aided INS** and **INS-aided GPS**. In the first approach, additional measurements provided by GPS are used to aid the INS for navigation by e.g., correcting the INS navigation errors due to the gyro drift and accelerometer bias. In the second approach, the INS measurements are used to improve the GPS positioning accuracy by estimating the estimation errors caused by the various measurement errors (and possibly poor GDOP). These two approaches clearly rely more on the integrity of the INS and GPS, respectively. Since INS is superior to GPS in terms of self-containment, it is more natural and advantageous to have a GPS-aided INS for navigation in most applications.

It is worthwhile to note that integrated navigation is in fact a special problem of data fusion and thus all effective data fusion techniques are at least potentially applicable to integrated navigation. The **data fusion** problem can be loosely stated as how to best extract useful information from multiple sets of data from *different sources* that are available. Since navigation is concerned mainly with the estimation of the state of the vehicle, the most relevant results in data fusion are those dealing with estimation.

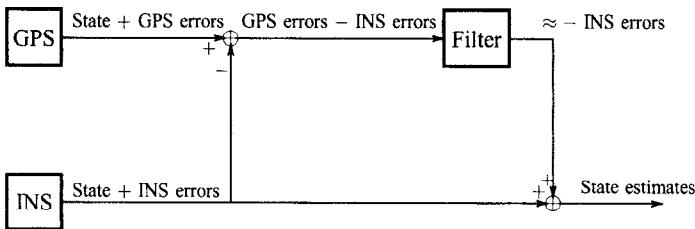
In this section, the following data fusion approaches (configurations or architectures) to INS/GPS integration are briefly discussed (for more details, see [Bar-Shalom95], Chapter 8):

- Complementary filtering
- Centralized estimation fusion
- Distributed estimation fusion

### 12.10.1 Integration by Complementary Filtering

The most commonly used form of INS/GPS integration in practice is based on complementary filtering, as described in Section 12.2. A conceptual diagram of such integration is given in Fig. 12.10.1-1.

It is clear from a comparison of Fig. 12.10.1-1 and Fig. 12.2.2-1 that INS and GPS provide two independent sets of measurements of the vehicle's state process, and the state estimator acts as the complementary filter. This filter,



**Figure 12.10.1-1:** Open-loop configuration for INS/GPS integration.

which operates on the difference of the errors, will reduce the GPS errors, resulting in its output being (approximately) the negative of the INS errors, which then added to the output of the INS, cancel (approximately) the INS errors, yielding the state estimates.

This is an *open-loop* (or feedforward) configuration. The state estimator is nonlinear since nonlinearities are involved in, e.g., the GPS measurements. This configuration provides a linearization mechanism particularly suitable and natural for the use of a linearized (or perturbed) Kalman filter: The output of the INS forms a reference/nominal trajectory, and the Kalman filter using the GPS measurement estimates the error of this nominal trajectory.

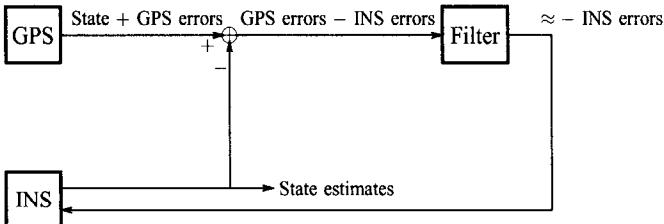
Besides, complementary filtering provides a distortionless processing of the signal, as explained before. This is particularly beneficial in the case where a rapid filter response is desirable since the filter operates only on the errors rather than on the full state.

Alternatively, a *closed-loop* (or feedback) configuration can be used, as shown in Fig. 12.10.1-2, where the inertial error estimates are fed back into the INS to reduce the error of the INS output on which the state estimator operates. The corresponding linearization mechanism is in the same spirit as that of the extended Kalman filter: linearizing about the latest estimates, in contrast to the linearized Kalman filter for the open-loop configuration. Whether closed- or open-loop configuration is superior depends primarily on how accurate the INS outputs are and thus they are, in general, better for long-term and short-term navigation applications, respectively.

## 12.10.2 Example

This example demonstrates how to use a Kalman filter to implement a complementary filtering configuration in an INS/GPS integration system.

The accelerometer errors will be neglected since they usually have a much smaller effect than the gyro errors. The error of each gyro is modeled by (12.4.2-1). Under some simplifying assumptions (e.g., the effect of platform



**Figure 12.10.1-2:** Closed-loop configuration for INS/GPS integration.

elevation angle error is negligible) for a slow dynamics vehicle (a ship), the platform tilt angle error vector  $[\phi_x, \phi_y, \phi_z]'$  in the (east, north, up) coordinate system is governed by, from the inertial navigation,

$$\begin{aligned}\phi_x &= \Omega \sin \varphi \phi_y - \Omega \cos \varphi \phi_z + \epsilon_x \\ \phi_y &= -\Omega \sin \varphi \phi_x + \epsilon_y \\ \phi_z &= \Omega \cos \varphi \phi_x + \epsilon_z\end{aligned}\quad (12.10.2-1)$$

where  $\Omega = 7.292115 \times 10^{-5}$  rad/s is the earth rotation rate and  $\varphi$  is the latitude of the ship. Combining this equation with (12.4.2-1) leads to the following state equation:

$$\dot{\mathbf{x}} = A\mathbf{x} + Bv \quad (12.10.2-2)$$

where  $\mathbf{x} = [\phi_x, \phi_y, \phi_z, \epsilon_{1,x}, \epsilon_{1,y}, \epsilon_{1,z}, \epsilon_{2,x}, \epsilon_{2,y}, \epsilon_{2,z}]'$  and

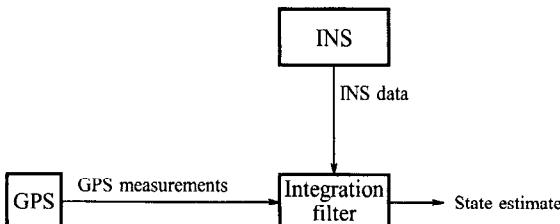
$$A = \begin{bmatrix} 0 & \Omega \sin \varphi & -\Omega \cos \varphi & I_{3 \times 3} & I_{3 \times 3} \\ -\Omega \cos \varphi & 0 & 0 & O_{3 \times 3} & O_{3 \times 3} \\ \Omega \cos \varphi & 0 & 0 & O_{3 \times 3} & diag(-\frac{1}{\tau_{\epsilon,x}}, -\frac{1}{\tau_{\epsilon,y}}, -\frac{1}{\tau_{\epsilon,z}}) \end{bmatrix} \quad (12.10.2-3)$$

$$B = \begin{bmatrix} O_{3 \times 3} \\ I_{3 \times 3} \\ O_{3 \times 3} \end{bmatrix} \quad (12.10.2-4)$$

$$v = [v_{\epsilon,x}, v_{\epsilon,y}, v_{\epsilon,z}]' \quad (12.10.2-5)$$

As shown in Fig. 12.10.1-1 or 12.10.1-2, the input to the integration filter is the difference between the GPS output and the INS output. The differences  $\Delta\varphi(t)$  and  $\Delta\lambda(t)$  between the GPS and INS measurements of the latitude and longitudinal of the ship, respectively, have the following relationship with  $\phi_x$  and  $\phi_y$ :

$$\begin{aligned}\Delta\varphi(t_k) &= \phi_x(t_k) + w_\varphi(t_k) \\ \Delta\lambda(t_k) &= -\phi_y(t_k)/\sin \varphi + w_\lambda(t_k)\end{aligned}\quad (12.10.2-6)$$



**Figure 12.10.3-1:** Tightly coupled configuration for INS/GPS integration.

where  $w_\varphi, w_\lambda$  are the GPS positioning errors. Assuming they are independent white noise, the measurement equation is

$$z_k = H\mathbf{x}_k + w_k \quad (12.10.2-7)$$

where  $z_k = [\Delta\varphi(t_k), \Delta\lambda(t_k)]'$ ,  $R_k = \text{cov}(w_k) = \text{diag}(\sigma_\varphi^2, \sigma_\lambda^2)$ , and

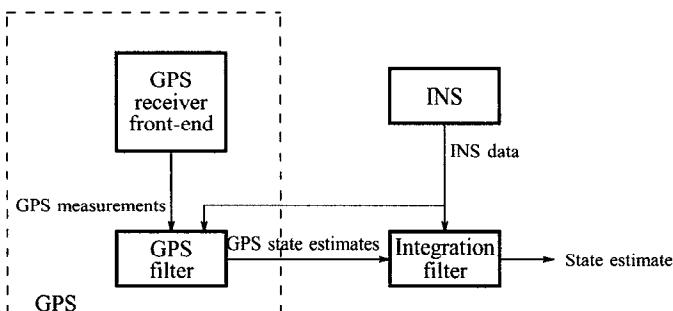
$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1/\sin\varphi & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (12.10.2-8)$$

A Kalman filter can then be implemented based on the above dynamic and measurement equations. Its output yields the estimates of the INS errors. The final state estimates are the INS system output with corrections by these estimates, as in the open-loop configuration of Fig. 12.10.1-1, or the output of the INS system where the errors are compensated by these error estimates internally, as in the closed-loop configuration of Fig. 12.10.1-2.

### 12.10.3 Integration by Centralized Estimation Fusion

A straightforward and theoretically better way of integrating INS and GPS is to employ an optimal filter that utilizes all measurements from both INS and GPS. This is known as **centralized estimation** in data fusion and **tightly coupled** configuration in INS/GPS integration, as illustrated in Fig. 12.10.3-1. In this configuration, both the GPS and INS measurements (e.g., pseudorange and delta range) are fed into the integration filter. The major advantage of this configuration lies in the flexibility of the integration filter in processing the uncorrelated GPS data from each satellite individually.

The major disadvantages of such integration include a heavy computational burden and poor fault-tolerance. The state in the state-space model would have to include the state components for the error processes of both inertial sensors and GPS receivers, as well as all kinematic state components of the vehicle. This leads to a high-dimensional filter with a heavy computational burden. Furthermore, the filter is vulnerable to failures of any part of the INS and GPS.



**Figure 12.10.4-1:** Loosely coupled configuration for INS/GPS integration.

In addition, this centralized architecture for integration is more complicated than some other architectures due to, e.g., the fact that INS and GPS have different data rates.

For improved fault tolerance, it is better for the integration filter to process the measurements one by one in a sequential fashion with a built-in mechanism of detecting and ignoring the faulty measurements.

#### 12.10.4 Integration by Distributed Estimation Fusion

In addition to the tightly coupled configuration, the integration of an INS and GPS can also be *loosely coupled*, as illustrated in Fig. 12.10.4-1. The loosely coupled configuration is also known as *cascaded filtering* since it involves two filters in cascade. In this configuration, the output (estimate) of the first filter within the GPS receiver and the INS (unfiltered) data are fed into the integration filter. This is a simpler and more economical approach since most GPS receivers already have a built-in Kalman filter. Also, its integration filter has a reduced dimension because it does not have to include e.g., the clock bias and drift as state components. However, its performance is inferior to the tightly coupled configuration primarily due to its slower rate of signal exchange and significant time delay. This configuration can be treated as a special case of a class of architecture known as *distributed estimation fusion*.

The basic idea of *INS/GPS integration by distributed estimation fusion* is as follows. Two filters run *in parallel* for INS and GPS, respectively. The final platform state is obtained by the integration filter by combining/fusing the estimates from these two filters using the proper fusion equations. For this purpose, some simple fusion formulas are presented next.

## Fusion of Estimates with Uncorrelated Errors

Assume  $\hat{x}_1, P_1$  and  $\hat{x}_2, P_2$  are the optimal linear estimates of the same parameter or state  $x$  and their associated covariance matrices from two estimators using two different measurement sets  $Z_1$  and  $Z_2$ . If the estimation errors associated with  $\hat{x}_1$  and  $\hat{x}_2$  are *uncorrelated*, then the optimal linear estimator of using measurement sets  $Z_1$  and  $Z_2$  can be shown to be

$$\hat{x} = P(P_1^{-1}\hat{x}_1 + P_2^{-1}\hat{x}_2) \quad (12.10.4-1)$$

$$P = (P_1^{-1} + P_2^{-1})^{-1} \quad (12.10.4-2)$$

Such equations are known as the *fusion equations*. Applying this to the fusion of INS and GPS filter estimates, one has the integration filter, at any time of integration,

$$\hat{x} = P[P_{\text{INS}}^{-1}\hat{x}_{\text{INS}} + P_{\text{GPS}}^{-1}\hat{x}_{\text{GPS}}] \quad (12.10.4-3)$$

$$P = [P_{\text{INS}}^{-1} + P_{\text{GPS}}^{-1}]^{-1} \quad (12.10.4-4)$$

## Is There Such a Thing as Estimates with Uncorrelated Errors?

The above formulas are correct only if the two estimates have uncorrelated errors. When one estimates the same state, which evolves according to a dynamic equation that includes process noise, with measurements from different sensors (with independent measurement noises), the corresponding estimation errors will be correlated because both are affected by the same process noise. Thus the above equations are clearly not optimal in practice.

## A Practical Exact Fusion Algorithm

If both INS and GPS systems transmit to the integration filter their estimates and predictions *at each time an update is made*, one can reconstruct the optimal fused estimate (the one that would be obtained by the centralized estimator) as follows.

Let  $\hat{x}_i(k|k-1), P_i(k|k-1)$  and  $\hat{x}_i(k|k), P_i(k|k)$  be the predicted and updated state estimates and covariances of filter  $i$  (either INS or GPS filter), respectively, and let  $\hat{x}(k|k-1), P(k|k-1)$  be the predicted state estimates and covariances of the integration filter. The optimal (conditional mean) fused estimate can be shown to be, for linear observations with Gaussian random variables ([Bar-Shalom95] Section 8.6),

$$\begin{aligned} \hat{x}(k|k) &= P(k|k) \left\{ P(k|k-1)^{-1}\hat{x}(k|k-1) \right. \\ &\quad \left. + [P_{\text{INS}}(k|k)^{-1}\hat{x}_{\text{INS}}(k|k) - P_{\text{INS}}(k|k-1)^{-1}\hat{x}_{\text{INS}}(k|k-1)] \right\} \end{aligned}$$

$$+ [P_{\text{GPS}}(k|k)^{-1} \hat{x}_{\text{GPS}}(k|k) - P_{\text{GPS}}(k|k-1)^{-1} \hat{x}_{\text{GPS}}(k|k-1)] \} \\ (12.10.4-5)$$

$$\begin{aligned} P(k|k) = & \left\{ P(k|k-1)^{-1} + P_{\text{INS}}(k|k)^{-1} - P_{\text{INS}}(k|k-1)^{-1} \right. \\ & \left. + P_{\text{GPS}}(k|k)^{-1} - P_{\text{GPS}}(k|k-1)^{-1} \right\}^{-1} \end{aligned} \quad (12.10.4-6)$$

since the measurement errors for the local filters are independent given the true state.

When applying the above fusion results to INS/GPS integration, the following two points deserve attention. First, although the INS filter usually estimates only the errors of the INS outputs, fusion of the INS and GPS should be done for the estimates of the full state (i.e., the error state estimates plus the INS outputs). Second, the filters for INS and GPS have different state components — e.g., the GPS filter usually does not yield an estimate of the vehicle’s attitude directly — and thus only the position and velocity estimates from the INS and GPS filters should be fused.

The above distributed estimation techniques can be made more fault-tolerant in the sense that the integrated estimates are still good even if a failure occurs to some local filters or sensors. A suboptimal approach to this is the so-called “federated filtering” [Carlson90, Gao93].

## 12.11 NOTES AND PROBLEMS

### 12.11.1 Bibliographical Notes

A number of books are available on navigation. For the general principles of land, sea, air, and space navigation without going into much mathematics, see [Kayton90]. The INS and GPS are the topics of many publications, such as [Broxmeyer64, Pitman62, Parkinson96]. The special issue [Engel99] contains much of the principles and latest advances in GPS and its major application areas. A book on applied Kalman filtering that includes navigation application is [Brown97]. A good and brief coverage of complementary filtering can be found in [Brown72]. [Liggins97] contains a very brief summary of important results in distributed estimation.

### 12.11.2 Problems

- 12-1 Covariance of INS model errors.** Find the covariance  $Q$  of the process noise  $v(k)$  in the state equation (12.4.3-8) that is discretized from the continuous-time model (12.4.3-1)–(12.4.3-6). Use the following approximation when necessary:

$$e^{A(T-\tau)} \approx I + A(T-\tau)$$

**12-2 Geometric Dilution of Precision in GPS.** Consider the following simplified GPS problem. There are four satellites in a vertical plane at locations given in polar coordinates as  $(r, \alpha)$ ,  $(r, \pi - \alpha)$ ,  $(r, \beta)$ ,  $(r, \pi - \beta)$ , where  $0 < \alpha < \beta < \pi/2$ . The location of a platform (situated at the origin) is to be estimated based on four range measurements from these satellites, with additive zero-mean i.i.d. noises with variance  $\sigma_r^2$ .

1. Plot the GDOP (total position RMS error divided by  $\sigma_r$ ) as a function of  $\alpha$  and  $\beta$ .
2. Find analytically the minimum GDOP.
3. If  $\alpha = 0$  and  $\beta = \pi/3$ , i.e., the satellites are “evenly” spaced, compare the vertical error with the horizontal one in the estimation of the location of the platform as above. Which is larger and why?

**12-3 Covariance of receiver block state noise.**

1. Derive (12.8.1-4).
2. Is the clock bias  $b$  modeled having a nearly constant changing rate (i.e., nearly constant clock drift  $d$ )? Compared with the nearly constant velocity model, which has a noise covariance given by (6.2.2-12), why is  $Q_c$  given by (12.8.1-4) rather than by  $Q_c = S_d V_2$ ?
3. Find a dynamic model of the clock state  $x_c$  that is the *direct discrete-time* counterpart of the model given in Subsection 12.8.1.

**12-4 Dynamic models for navigation with GPS.** The discrete-time dynamic model in the state space described in Subsection 12.8.2 is obtained by discretizing the corresponding continuous-time model and is valid only for a vehicle with a nearly constant velocity. In the following, include the clock state as part of the state vector.

1. Find the dynamic model of the state that is the *direct discrete-time* counterpart of the model given in Subsection 12.8.2.
2. Find a continuous-time dynamic model in the state space and its discretized version that are valid for a vehicle with a nearly constant acceleration.
3. Find a dynamic model that is the *direct discrete-time* counterpart of the model developed in part 2 of this problem.
4. Find the dynamic model for a vehicle with a nearly coordinated turn motion.

**12-5 State-space model for CV motion with GPS measurements in the presence of SA.** Write down the discrete-time vector dynamic equation and linearized measurement equation for the extended Kalman filter for a vehicle at nearly constant velocity based on the GPS measurements of four satellites in the presence of Selective Availability. Use the first-order Markov model of Subsection 12.8.4 to model SA.

**12-6 Mobile sensor platform location error.** The position of a moving sensor platform is estimated with GPS at a frequency of 1 Hz. Because the GPS measurements are filtered, the output of such a filter — the estimated location of the platform — is characterized by autocorrelated errors (even under the ideal conditions when all the noises are white). When position measurements of remote targets are made with the on-board sensor (with zero mean measurement noises), the estimated locations of these targets are affected by the imperfectly known sensor location: The autocorrelated errors in the estimated

location of its platform act as a slowly varying bias on top of the sensor measurement noise. A model is therefore needed for these sensor location errors.

Assume the platform motion model as a direct discrete-time WNA model with zero-mean process noise with variance  $\sigma_v^2$  in a single dimension. Assume the (position) measurement noise has  $\sigma_w = 10 \text{ m}$ , with the usual assumptions.

1. Find the dynamic equation of the platform state estimation error  $\tilde{x}(k|k)$  driven by the process noise  $v$  and the measurement noise  $w$ .
2. Assuming the filter is in steady state, find the transfer function from these noises to the platform state estimation error.
3. Find the poles of the above transfer functions for  $\sigma_v = 0.1, 1, 10 \text{ m/s}^2$ .
4. Evaluate from these poles the (approximate) decorrelation time of the estimation error (the time it takes for the autocorrelation to decay by a factor of 10).
5. Give the exact expression of the (matrix) autocorrelation function of the (vector) estimation error.
6. Plot the exact autocorrelation function of the position estimation error for the above three cases. Compare with the approximate results from the previous item.

- 12-7 Mobile sensor platform location error reduction.** Even though the GPS based estimates of a platform location are very accurate, if the sensor is capable of high-accuracy range measurements, further refinement of the platform location accuracy can be obtained by making measurements on the (unknown) location of a target as long as this target is stationary.

1. The following range measurements (from the sensor to the target) are made:

$$r(k) = x_t(k) - x_p(k) + w_r(k) = x_t(k) - \hat{x}_p(k|k) - \tilde{x}_p(k|k) + w_r(k)$$

where  $x_p(k)$  is the true platform position,  $\hat{x}_p(k|k)$  is the estimate (available from GPS) with error  $\tilde{x}_p(k|k)$ , and  $w_r(k)$  is the sensor's range measurement noise. It is convenient to define a measurement that is shifted by a known quantity (the platform position estimate)

$$z(k) = r(k) + \hat{x}_p(k|k) = x_t(k) - \tilde{x}_p(k|k) + w_r(k)$$

Using the platform location error model from the previous problem (with state components, position  $\tilde{x}_p$  and velocity  $\tilde{x}_v$ ), formulate the simultaneous estimation of the composite state  $\mathbf{x}(k)$  consisting of  $\tilde{x}(k|k)$  and the target location  $x_t(k)$  (in one dimension) based on the measurements  $z(k)$ .

2. Is the resulting system completely observable?
3. Evaluate (using DynaEst) the steady-state RMS error in the platform position following these range measurements (which are in addition to the GPS) with  $\sigma_r = 1 \text{ m}$  for the three process noise levels from the previous problem.

- 12-8 Effect of satellite timing errors on positioning accuracy.** Consider the case where four satellites are visible. The pseudorange equation (12.6.3-1) can be rewritten as

$$\begin{aligned} f_1 &= (X_1 - x)^2 + (Y_1 - y)^2 + (Z_1 - z)^2 - c^2(t_1 + \tau)^2 \\ f_2 &= (X_2 - x)^2 + (Y_2 - y)^2 + (Z_2 - z)^2 - c^2(t_2 + \tau)^2 \\ f_3 &= (X_3 - x)^2 + (Y_3 - y)^2 + (Z_3 - z)^2 - c^2(t_3 + \tau)^2 \\ f_4 &= (X_4 - x)^2 + (Y_4 - y)^2 + (Z_4 - z)^2 - c^2(t_4 + \tau)^2 \end{aligned}$$

where  $\tau$  is the user clock offset from the GPS standard time, and  $t_i$  is the  $i$ th satellite's time offset from the GPS standard time. The positioning accuracy along  $x$  axis in terms of variation in  $t_i$  is given approximately by

$$\Delta x \approx \sum_i \frac{\partial x}{\partial t_i} \Delta t_i$$

In the above,  $\frac{\partial x}{\partial t_i}$  can be found by the implicit function theorem:

$$\frac{\partial \mathbf{x}}{\partial \mathbf{t}} = - \left( \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{t}}$$

where  $\mathbf{f} = [f_1, f_2, f_3, f_4]', \mathbf{x} = [x, y, z, \tau]', \mathbf{t} = [t_1, t_2, t_3, t_4]'$ . If  $|\Delta t_i| \leq \Delta T$  for all  $i$ , then

$$|\Delta x| \leq \Delta T \sum_i \left| \frac{\partial x}{\partial t_i} \right|$$

To have a positioning error greater than 100 m in the  $x$  coordinate, how large does  $\Delta T$  have to be at least? Such results can be used to figure out how large the "dither SA" may be. To have a 1 cm accuracy in positioning in the  $x$  coordinate, find the approximate upper bound on  $\Delta T$ . Use the following typical set of values (these values need to carry at least ten digits to ensure numerical accuracy) in  $10^6$  m and seconds:

$$\begin{array}{lll} x_1 = 1.876371950559744 & y_1 = -10.64143413406656 & z_1 = 24.2697646566144 \\ x_2 = 10.97666464137408 & y_2 = -13.08147952230029 & z_2 = 20.35116937827073 \\ x_3 = 24.58513954435968 & y_3 = -4.335023426659201 & z_3 = 9.08630032021747 \\ x_4 = 3.854136195752833 & y_4 = 7.248575943442946 & z_4 = 25.26630462778753 \\ t_1 = 0.07234683200 & t_2 = 0.06730845726 & t_3 = 0.06738499643 \\ t_4 = 0.07651971839 & c = 299792458.0 \text{ m/s} & \end{array}$$

### 12.11.3 Term Project — Extended Kalman Filter for GPS

Consider a user (an aircraft) equipped with a GPS receiver that has access to  $C/A$  code but not  $P$  code. At initial time  $t = 0$ , the user was at

$$\mathbf{x}_0 = [x \quad \dot{x} \quad y \quad \dot{y} \quad z]' = [0 \quad 0 \quad 0 \quad 120 \text{ m/s} \quad 0]'$$

where  $(x, y, z)$  is the coordinates of the aircraft in the locally level reference frame. The aircraft at beginning moves straight at a constant velocity  $[0 \quad 120 \text{ m/s} \quad 0]$ . It then completes a  $1^\circ/\text{s}$  "coordinated turn" of  $90^\circ$  in the horizontal  $x'-y'$  plane from due north to due east, starting at  $t = 500$  s. After it completes this turn, the aircraft continues moving straight with a constant velocity. The actual motion of the aircraft is the superposition of the motion described above and the random motion due to turbulence. The turbulence can be modeled as a continuous-time zero-mean white Gaussian noise whose components represent the random and uncorrelated accelerations along  $x'$  and  $y'$  directions and velocity along  $z'$  direction, respectively. The true covariance matrix of the equivalent discrete-time noise is not known to the filter and is assumed to be

$$Q_t = \text{diag}[Q_2 \quad Q_2 \quad TS_z]$$

where  $Q_2 = S_w \begin{bmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{bmatrix}$ ,  $S_w = 0.01 \text{ m}^2/\text{s}^3$ , and  $S_z = 0.01 \text{ m}^2/\text{s}$ .

The GPS receiver equipped with a *crystal clock* receives synchronized signals from all visible satellites every one second (i.e., sampling interval  $T = 1$  s). For the pseudorange measurements, it is assumed that the measurement noise is Gaussian, white, zero mean and has a standard deviation of 30 m.

1. Following Subsection 12.9.1, generate the trajectories of all six satellites. Plot all six trajectories (overlay in one plot) in the locally level Cartesian coordinates  $(x', y', z')$  from 0 to 1200 s. Plot part of the spherical coordinates [i.e.,  $(\phi, \theta)$ , without the distance  $r$ ] of all six trajectories (overlay in one plot) in the locally level coordinate system from  $t = 1$  to  $t = 1200$  s.
2. Generate the true state trajectory of the aircraft in the local coordinates. Plot a single (i.e., one run) three-dimensional positional trajectory.
3. Plot (in one plot) the six noisy pseudoranges between the aircraft and the six satellites for a single run.
4. Design and implement an extended Kalman filter to estimate the three-dimensional position and velocity of this aircraft, *using the 7-state model* described in Subsection 12.9.2. Indicate the rationale for the choice of the estimator design parameters (e.g., noise statistics). Try to have as much as possible the “noise reduction” in the straight portions of the trajectory while maintaining the position RMS estimation errors not larger than the single measurement position RMS error during the turn. Explain explicitly how the filter was initialized and what initial estimate and covariance were used. *Give the final design parameters used.*
5. Test the extended Kalman filter assuming the GPS receiver receives signals from satellite sets
  - $\{1, 2, 3, 4, 5, 6\}$
  - $\{1, 2, 3, 4\}$
  - $\{1, 2, 4, 5\}$

respectively, for the period from  $t = 1$  to  $t = 1200$  s. Report the following from 100 Monte Carlo runs

- (a) (coordinate-combined) RMS position estimation errors (rms\_pos)
- (b) (coordinate-combined) RMS velocity estimation errors (rms\_vel)
- (c) (coordinate-combined) position error standard deviation (sd\_pos)
- (d) (coordinate-combined) velocity error standard deviation (sd\_vel)
- (e) Average normalized estimation error squared (NEES)
- (f) Average normalized error (ave\_norm\_err) of the first component  $\frac{\hat{x}_1(k|k)}{\sqrt{P_{11}(k|k)}}$
- (g) The sequences of GDOPs (use a semilog plot)
- (h) Compare rms\_pos with sd\_pos and rms\_vel with sd\_vel

Overlay the error results (not the GDOP results) for different satellite sets on one plot, i.e., have a single plot for each of the parts (a) through (f).

Provide the expressions you used to calculate rms\_pos, rms\_vel, sd\_pos, and sd\_vel. Adjust the scale of your plots such that the variation of most portions of the curves is visible. Every plot should be numbered with a proper caption.

6. Comment on the accuracy of the extended Kalman filter when different sets of satellites are visible.

**Deliverables:**

1. A hardcopy of a concise report, including items listed above, all necessary formulas (with explanations), important results (in figures or tables) to support your findings or conclusions, and discussion of the results.
2. A floppy disk containing the main routine and all subroutines with necessary documentation, and detailed results. Use two separate directories for routines and results.
3. A brief demo on a PC to show how your program works with some representative results.
4. Your program should be modularized as much as possible. For example, your main program should call an EKF subroutine (a function subroutine in MATLAB) ekf.m, which should in turn call a Kalman filter subroutine (a function subroutine in MATLAB) kf.m. All the subroutines should be generally valid for other cases (i.e., not limited to the current case).



# Bibliography

- [Ackerson70] G. A. Ackerson and K. S. Fu. On State Estimation in Switching Environments. *IEEE Trans. Automatic Control*, AC-15(1):10–17, Jan. 1970.
- [Aidala83] V. J. Aidala and S. E. Hammel. Utilization of Modified Polar Coordinates for Bearings-Only Tracking. *IEEE Trans. Automatic Control*, AC-28:283–293, March 1983.
- [Anderson79] B. O. D. Anderson and J.B. Moore. *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, NJ, 1979.
- [Andrisani86] D. Andrisani, F. P. Kuhl, and D. Gleason. A Nonlinear Tracker Using Attitude Measurements. *IEEE Trans. Aerospace and Electronic Systems*, AES-22:533–539, Sept. 1986.
- [Athans68] M. Athans, R. P. Wishner, and A. Bertolini. Suboptimal State Estimation for Continuous-Time Nonlinear Systems from Discrete Noisy Measurements. *IEEE Trans. Automatic Control*, AC-13:504–514, Oct. 1968.
- [Baheti90] R. S. Baheti, A. J. Laub, and D. M. Wiberg (Eds.). Bierman Memorial Issue on Factorized Estimation Applications. *IEEE Trans. Automatic Control*, AC-35:1282–1319, Dec. 1990.
- [Balakrishnan84] A. V. Balakrishnan. *Kalman Filtering Theory*. University Series in Modern Engineering, Springer, Berlin, Germany, 1984.
- [Bar-Shalom82] Y. Bar-Shalom and K. Birmiwal. Variable Dimension Filter for Maneuvering Target Tracking. *IEEE Trans. Aerospace and Electronic Systems*, AES-18(5):621–629, Sept. 1982.
- [Bar-Shalom83] Y. Bar-Shalom and K. Birmiwal. Consistency and Robustness of PDAF for Target Tracking in Cluttered Environments. *Automatica*, 19:431–437, July 1983.
- [Bar-Shalom88] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, New York, 1988.
- [Bar-Shalom89] Y. Bar-Shalom, K. C. Chang, and H. A. P. Blom. Tracking a Maneuvering Target Using Input Estimation vs. the Interacting Multiple Model Algorithm. *IEEE Trans. Aerospace and Electronic Systems*, AES-25:296–300, March 1989.
- [Bar-Shalom95] Y. Bar-Shalom and X. R. Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS Publishing, 1995.
- [Bar-Shalom98a] Y. Bar-Shalom (Ed.). *Multitarget-Multisensor Tracking: Advanced Applications*. Artech House, Norwood, MA, 1990. Reprinted by YBS Publishing, 1998.
- [Bar-Shalom98b] Y. Bar-Shalom (Ed.). *Multitarget-Multisensor Tracking: Applications and Advances*, Vol. II. Artech House, Norwood, MA, 1992. Reprinted by YBS Publishing, 1998.
- [Baram78] Y. Baram and N. R. Sandell. Consistent Estimation on Finite Parameter Sets with Application to Linear Systems Identification. *IEEE Trans. Automatic Control*, AC-23(3):451–454, June 1978.
- [Barniv85] Y. Barniv. Dynamic Programming Solution to Detecting Dim Moving Targets. *IEEE Trans. Aerospace and Electronic Systems*, AES-21:144–156, Jan. 1985.
- [Barret90] I. Barret. *Synthese d'algorithmes de Poursuite Multi-Radar d'avions Civils Manoeuvrants*. PhD thesis, Ecole Nationale Supérieure de l'aéronautique et de l'espace, Toulouse, France, 1990.

## 538 BIBLIOGRAPHY

- [Bellman61] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, Princeton, NJ, 1961.
- [Benedict62] T. R. Benedict and G. W. Bordner. Synthesis of an Optimal Set of Radar Track-While-Scan Smoothing Equations. *IRE Trans. Automatic Control*, AC-7:27–32, July 1962.
- [Bertsekas92] D. P. Bertsekas. *Linear Network Optimization: Algorithms and Codes*. The MIT Press, Cambridge, MA, 1992.
- [Bethel87] R. E. Bethel and R. G. Rahikka. An Optimum First-Order Time Delay Tracker. *IEEE Trans. Aerospace and Electronic Systems*, AES-23:718–725, Nov. 1987.
- [Bierman77] G. J. Bierman. *Factorization Methods for Discrete Sequential Estimation*. Academic Press, New York, 1977.
- [Blackman81] S. S. Blackman, T. J. Broida, and M. F. Cartier. Applications of a Phased Array Antenna in a Multiple Maneuvering Target Environment. In *Proc. 20th IEEE Conf. Decision and Control*, San Diego, CA, Dec. 1981.
- [Blackman86] S. S. Blackman. *Multiple Target Tracking with Radar Applications*. Artech House, Norwood, MA, 1986.
- [Blackman99] S. S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, Norwood, MA, 1999.
- [Blair98] W. D. Blair, G. A. Watson, T. Kirubarajan, and Y. Bar-Shalom. Benchmark for Radar Resource Allocation and Tracking Targets in the Presence of ECM. *IEEE Trans. Aerospace and Electronic Systems*, AES-34, Oct. 1998.
- [Blom84] H. A. P. Blom. An Efficient Filter for Abruptly Changing Systems. In *Proc. 23rd IEEE Conf. Decision and Control*, Las Vegas, NV, Dec. 1984.
- [Blom85] H. A. P. Blom. An Efficient Decision-Making-Free Filter for Processes with Abrupt Changes. In *Proc. 7th IFAC Symp. Identif. and System Parameter Estimation*, York, U.K., July 1985.
- [Blom88] H. A. P. Blom and Y. Bar-Shalom. The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients. *IEEE Trans. Automatic Control*, AC-33(8):780–783, Aug. 1988.
- [Bogler87] P. L. Bogler. Tracking a Maneuvering Target Using Input Estimation. *IEEE Trans. Aerospace and Electronic Systems*, AES-23(3):298–310, May 1987.
- [Braasch90] M. S. Braasch. A Signal Model for GPS. *Navigation: Journal of The Institute of Navigation*, 37(4), Winter 1990.
- [Brookner98] E. Brookner. *Tracking and Kalman Filtering Made Easy*. John Wiley & Sons, New York, 1998.
- [Brown72] R. G. Brown. Integrated Navigation Systems and Kalman Filtering: A Perspective. *Navigation: Journal of The Institute of Navigation*, 19(4), Winter 1972.
- [Brown97] R. G. Brown and P. Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, New York, 1997.
- [Broxmeyer64] C. Broxmeyer. *Inertial Navigation Systems*. McGraw-Hill, New York, 1964.
- [Bryson75] A. Bryson and Y. C. Ho. *Applied Optimal Control*. Hemisphere, Cambridge, MA, 1975.
- [Caglayan80] A. K. Caglayan. Simultaneous Failure Detection and Estimation in Linear Systems. In *Proc. 19th IEEE Conf. Decision and Control*, Albuquerque, NM, Dec. 1980.
- [Carlson90] N. A. Carlson. Federated Square Root Filter for Decentralized Parallel Proccses. *IEEE Trans. AES*, 26(3), May 1990.
- [Carvalho97] H. Carvalho, P. Del Moral, A. Monin, and G. Salut. Optimal Nonlinear Filtering in GPS/INS Integration. *IEEE Trans. AES*, 33(3), July 1997.
- [Castella80] F. R. Castella. An Adaptive Two-Dimensional Kalman Tracking Filter. *IEEE Trans. Aerospace and Electronic Systems*, AES-16:822–829, Nov. 1980.

- [Castella81] F. R. Castella. Tracking Accuracies with Position and Rate Measurements. *IEEE Trans. Aerospace and Electronic Systems*, AES-17:433–437, May 1981.
- [Chan79] Y. T. Chan, A. G. C. Hu, and J. B. Plant. A Kalman Filter Based Tracking Scheme with Input Estimation. *IEEE Trans. Aerospace and Electronic Systems*, AES-15(2):237–244, March 1979.
- [Chang77] C. B. Chang, R. H. Whiting, and M. Athans. On the State and Parameter Estimation for Maneuvering Reentry Vehicles. *IEEE Trans. Automatic Control*, AC-22(2):99–105, Feb. 1977.
- [Chang78] C. B. Chang and M. Athans. State Estimation for Discrete Systems with Switching Parameters. *IEEE Trans. Aerospace and Electronic Systems*, AES-14(5):418–425, May 1978.
- [Chang84] C. B. Chang, K. P. Dunn, and L. C. Youens. A Tracking Algorithm for Dense Target Environments. In *Proc. 1984 American Control Conf.*, San Diego, CA, June 1984.
- [Chen84] C. T. Chen. *Linear System Theory and Design*. Holt, New York, 1984.
- [Cortina91] E. Cortina, D. Otero, and C. E. D'Attellis. Maneuvering Target Tracking Using EKF. *IEEE Trans. Aerospace and Electronic Systems*, AES-27:155–158, Jan. 1991.
- [Daum88] F. E. Daum. New Exact Nonlinear Filters. In J. C. Spall (Ed.), *Bayesian Analysis of Time Series and Dynamic Models*. Dekker, New York, 1988.
- [Daum94] F. E. Daum. New exact nonlinear filters: Theory and applications. In *Proc. SPIE Conf.*, Vol. 2235, pages 636–649, Orlando, FL, April 1994.
- [Daum95] F. E. Daum. Beyond Kalman Filters: Practical Design of Nonlinear Filters. In *SPIE Proceedings*, Vol. 2561, July 1995.
- [Demirbas87] K. Demirbas. Maneuvering Target Tracking with Hypothesis Testing. *IEEE Trans. Aerospace and Electronic Systems*, AES-23:757–766, Nov. 1987.
- [Doucet00] A. Doucet, S. Godsill, and C. Andrieu. On Sequential Simulation-Based Methods for Bayesian Filtering. *Statistics and Computing*, 10(3):197–208, April 2000.
- [Doucet01] A. Doucet, J. F. G. de Freitas, and N. J. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001.
- [Dougherty95] J. J. Dougherty, H. El-Shaerief, D. J. Simon, and G. A. Whitmer. GPS Modeling for Designing Aerospace Vehicle Navigation Systems. *IEEE Trans. AES*, 31(2), April 1995.
- [Drummond93] O. E. Drummond and G. Frenkel. Glossary of Tracking Terms of SDI Panels on Tracking. In *Proc. SPIE Conf. on Signal and Data Processing of Small Targets*, Vol. 1954, April 1993.
- [Ekstrand83] B. Ekstrand. Analytical Steady State Solution for a Kalman Tracking Filter. *IEEE Trans. Aerospace and Electronic Systems*, AES-19:815–819, Nov. 1983.
- [Ekstrand85] B. Ekstrand. Analytical Steady-State Solution for a Continuous Time Kalman Filter. *IEEE Trans. Aerospace and Electronic Systems*, AES-21:746–750, Nov. 1985.
- [Enge99] P. Enge and P. Misra (Eds.). Global Positioning System, Special Issue. *Proceedings of IEEE*, 87(1), 1999.
- [Farina85] A. Farina and F. A. Studer. *Radar Data Processing*, Vol. I: *Introduction and Tracking*, Vol. II: *Advanced Topics and Applications*. Research Studies Press, Letchworth, Hertfordshire, England, 1985.
- [Fitzgerald74] R. J. Fitzgerald. Effects of Range-Doppler Coupling on Chirp Radar Tracking Accuracy. *IEEE Trans. Aerospace and Electronic Systems*, AES-10:528–532, July 1974.
- [Fitzgerald80] R. J. Fitzgerald. Simple Tracking Filters: Filtering and Smoothing Performance. *IEEE Trans. Aerospace and Electronic Systems*, AES-16:860–864, Nov. 1980.
- [Fitzgerald81] R. J. Fitzgerald. Simple Tracking Filters: Closed Form Solutions. *IEEE Trans. Aerospace and Electronic Systems*, AES-17:781–785, Nov. 1981.
- [Fletcher87] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, Chichester, England, 1987.

## 540 BIBLIOGRAPHY

- [Fogel88] E. Fogel and M. Gavish. Nth-order dynamics target observability from angle measurements. *IEEE Trans. Aerospace and Electronic Systems*, AES-24(3):305–308, May 1988.
- [Fortmann77] T. E. Fortmann and K. L. Hitz. *An Introduction to Linear Control Systems*. Dekker, New York, 1977.
- [Fraser69] D. C. Fraser and J. E. Potter. The Optimum Smoother as a Combination of Two Optimum Linear Filters. *IEEE Trans. Automatic Control*, AC-14(4):387–390, August 1969.
- [Friedland69] B. Friedland. Treatment of Bias in Recursive Filtering. *IEEE Trans. Automatic Control*, AC-14:359–367, Aug. 1969.
- [Friedland73] B. Friedland. Optimum Steady State Position and Velocity Estimation Using Noisy Sampled Position Data. *IEEE Trans. Aerospace and Electronic Systems*, AES-9:906–911, Nov. 1973.
- [Galkowski91] P. J. Galkowski and M. A. Islam. An Alternative Derivation of the Modified Gain Function of Song and Speyer. *IEEE Trans. Automatic Control*, 36:1323–1326, Nov. 1991.
- [Gao93] Y. Gao, E. J. Krakiwsky, and M. A. Abousalem. Comparison and Analysis of Centralized, Decentralized, and Federated Filters. *Navigation: Journal of The Institute of Navigation*, 40(1), Spring 1993.
- [Gelb74] A. Gelb. *Applied Optimal Estimation*. MIT Press, Cambridge, MA, 1974.
- [Gholson77] N. H. Gholson and R. L. Moose. Maneuvering Target Tracking Using Adaptive State Estimation. *IEEE Trans. Aerospace and Electronic Systems*, AES-13:310–317, May 1977.
- [Gordon93] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. A Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. *IEE Proceedings on Radar and Signal Processing*, 140:107–113, July 1993.
- [Gray93] J. Gray and W. Murray. A Derivation of an Analytic Expression for the Alpha-Beta-Gamma Filter. *IEEE Trans. Aerospace and Electronic Systems*, AES-29:1064–1065, July 1993.
- [Grewal93] M. S. Grewal and A. P. Andrews. *Kalman Filtering: Theory and Practice*. Prentice-Hall, Englewood Cliffs, NJ, 1993.
- [Hadidi78] M. T. Hadidi and S. C. Schwartz. Sequential Detection with Markov Interrupted Observations. In *Proc. 16th Allerton Conf. on Communication, Control and Computing*, Univ. of Illinois, Oct. 1978.
- [Ham83] F. M. Ham and R. G. Brown. Observability, Eigenvalues and Kalman Filtering. *IEEE Trans. Aerospace and Electronic Systems*, AES-34(3):269–273, Mar. 1983.
- [Hampton73] R. L. T. Hampton and J. R. Cooke. Unsupervised Tracking of Maneuvering Vehicles. *IEEE Trans. Aerospace and Electronic Systems*, AES-9:197–207, March 1973.
- [Hanlon98] P. D. Hanlon and P. S. Maybeck. Interrelationship of Single-Filter and Multiple-Model Adaptive Algorithms. *IEEE Trans. Aerospace and Electronic Systems*, AES-34:934–946, July 1998.
- [Helmick95] R. E. Helmick, W. D. Blair, and S. A. Hoffman. Fixed-Interval Smoothing for Markovian Switching Systems. *IEEE Trans. Information Theory*, 41:1845–1855, Nov. 1995.
- [Itzkowitz89] H. R. Itzkowitz and R. S. Baheti. Demonstration of Square Root Kalman Filter on WARP Parallel Computer. In *Proc. 1989 American Control Conf.*, Pittsburgh, PA, June 1989.
- [Jaarsma91] D. Jaarsma and R. A. Davis. Joint Estimation of Time Delay and Doppler Ratio from Broadband Using Discrete Estimation Techniques. In *Proc. 1991 International Conf. Acoustics, Speech, and Signal Processing*, Toronto, Canada, May 1991.
- [Jaffer71a] A. G. Jaffer and S. C. Gupta. Optimal Sequential Estimation of Discrete Processes with Markov Interrupted Observations. *IEEE Trans. Automatic Control*, AC-16:417–475, Oct. 1971.
- [Jaffer71b] A. G. Jaffer and S. C. Gupta. Recursive Bayesian Estimation with Uncertain Observation. *IEEE Trans. Information Theory*, IT-17:614–616, Sept. 1971.
- [Jazwinski69] A. H. Jazwinski. Adaptive Filtering. *Automatica*, 5(4):475–485, July 1969.

- [Jazwinski70] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970.
- [Johnston72] J. Johnston. *Econometric Methods*. McGraw-Hill, New York, 1972.
- [Kailath81] T. Kailath. *Lectures on Wiener and Kalman Filtering*. Springer-Verlag, New York, 1981.
- [Kalata84] P. R. Kalata. The Tracking Index: A Generalized Parameter for Alpha-Beta-Gamma Target Trackers. *IEEE Trans. Aerospace and Electronic Systems*, AES-20:174–182, March 1984.
- [Kalman60] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Trans. ASME, J. Basic Engineering*, 82:34–45, March 1960.
- [Kalman61] R. E. Kalman and R. Bucy. New Results in Linear Filtering and Prediction Theory. *Trans. ASME, J. Basic Engineering*, 83:95–108, March 1961.
- [Kayton90] M. Kayton (Ed.). *Navigation: Land, Sea, Air, and Space*. IEEE Press, New York, 1990.
- [Kendrick81] J. D. Kendrick, P. S. Maybeck, and J. G. Reid. Estimation of Aircraft Target Motion Using Orientation Measurements. *IEEE Trans. Aerospace and Electronic Systems*, 17:254–260, March 1981.
- [Kerr89a] T. H. Kerr. Duality Between Failure Detection and Radar/Optical Maneuver Detection. *IEEE Trans. Aerospace and Electronic Systems*, AES-25:520–528, July 1989.
- [Kerr89b] T. H. Kerr. Status of CR-like Lower Bounds for Nonlinear Filtering. *IEEE Trans. Aerospace and Electronic Systems*, AES-25:590–601, September 1989.
- [Kirubarajan96] T. Kirubarajan and Y. Bar-Shalom. Target Motion Analysis in Clutter for Passive Sonar Using Amplitude Information. *IEEE Trans. Aerospace and Electronic Systems*, AES-32:1367–1384, Oct. 1996.
- [Kirubarajan98] T. Kirubarajan, Y. Bar-Shalom, W. D. Blair, and G. A. Watson. IMMPDA Solution to Benchmark for Radar Resource Allocation and Tracking Targets in the Presence of ECM. *IEEE Trans. Aerospace and Electronic Systems*, AES-34:1115–1134, Oct. 1998.
- [Kirubarajan00] T. Kirubarajan and Y. Bar-Shalom. Kalman Filter vs. IMM Estimator: When Do We Need the Latter? In *Proc. SPIE Conf. Signal Processing, Sensor Fusion and Target Recognition (#4048)*, Orlando, FL, April 2000.
- [Kleinman89] D. L. Kleinman. Private communication, 1989.
- [Korn82] J. Korn, S. W. Gully, and A. S. Willsky. Application of the Generalized Likelihood Ratio Algorithm to Maneuver Detection and Estimation. In *Proc. 1982 American Control Conf.*, Arlington, VA, June 1982.
- [Kramer87] S. C. Kramer and H. W. Sorenson. Bayesian Parameter Estimation. In *Proc. 1987 American Control Conf.*, Minneapolis, MN, June 1987.
- [Kramer88] S. C. Kramer and H. W. Sorenson. Recursive Bayesian Estimation Using Piece-wise Constant Approximations. *Automatica*, 24(6):789–801, Nov. 1988.
- [Larson66] R. E. Larson and J. Peschon. A Dynamic Programming Approach to Trajectory Estimation. *IEEE Trans. Automatic Control*, AC-11:537–540, July 1966.
- [Lefas84] C. C. Lefas. Using Roll-Angle Measurement to Track Aircraft Maneuvers. *IEEE Trans. Aerospace and Electronic Systems*, AES-20:672–681, Nov. 1984.
- [Lerro93] D. Lerro and Y. Bar-Shalom. Tracking with Debiased Consistent Converted Measurements vs. EKF. *IEEE Trans. Aerospace and Electronic Systems*, AES-29(3):1015–1022, July 1993.
- [Lewis86] F. L. Lewis. *Optimal Estimation*. John Wiley & Sons, New York, 1986.
- [Li93] X. R. Li and Y. Bar-Shalom. Performance Prediction of the Interacting Multiple Model Algorithm. *IEEE Trans. Aerospace and Electronic Systems*, AES-29(3):755–771, July 1993.
- [Li94] X. R. Li and Y. Bar-Shalom. A Recursive Multiple Model Approach to Noise Identification. *IEEE Trans. Aerospace and Electronic Systems*, AES-30(3):671–684, July 1994.

## 542 BIBLIOGRAPHY

- [Li95] X. R. Li and Y. Bar-Shalom. Performance Prediction of Hybrid Algorithms. In C. T. Leondes (Ed.), *Advances in Aerospace Systems Dynamics and Control Systems*, Vol. 72. Academic Press, New York, 1995.
- [Li96] X. R. Li. Hybrid Estimation Techniques. In C. T. Leondes (Ed.), *Advances in Aerospace Systems Dynamics and Control Systems*, Vol. 76. Academic Press, New York, 1996.
- [Li97] X. R. Li. Canonical Transformation for Tracking with Kinematic Models. *IEEE. Trans. Aerospace and Electronic Systems*, AES-33:1212–1224, October 1997.
- [Liggins97] M. Liggins, C. Y. Chong, I. Kadar, M. G. Alford, V. Vannicola, and S. Thomopoulos. Distributed Fusion Architectures and Algorithms for Target Tracking. *Proceedings of IEEE*, 85(1), 1997.
- [Lin01] X. Lin, T. Kirubarajan, Y. Bar-Shalom, and X. R. Li. Enhanced Accuracy GPS Navigation Using the Interacting Multiple Model Estimator. In *Proc. 2000 IEEE Aerospace Conf.*, Big Sky, MT, 2001.
- [Ljung87] L. Ljung. *System Identification*. Prentice-Hall, Englewood Cliffs, NJ, 1987.
- [Magill65] D. T. Magill. Optimal Adaptive Estimation of Sampled Stochastic Processes. *IEEE Trans. Automatic Control*, AC-10:434–439, 1965.
- [Maybeck79] P. S. Maybeck. *Stochastic Models, Estimation and Control*, Vol. I. Academic Press, New York, 1979.
- [Maybeck82] P. S. Maybeck. *Stochastic Models, Estimation and Control*, Vols. II and III. Academic Press, New York, 1982.
- [Mazor98] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan. Interacting Multiple Model Methods in Target Tracking: A Survey. *IEEE. Trans. Aerospace and Electronic Systems*, AES-34:103–123, Jan. 1998.
- [McAulay73] R. J. McAulay and E. J. Denlinger. A Decision-Directed Adaptive Tracker. *IEEE Trans. Aerospace and Electronic Systems*, AES-9(2):229–236, March 1973.
- [Meditch69] J. Meditch. *Stochastic Linear Estimation and Control*. McGraw-Hill, New York, 1969.
- [Mehra71] R. K. Mehra. A Comparison of Several Nonlinear Filters for Reentry Vehicle Tracking. *IEEE Trans. Automatic Control*, AC-16:307–319, Aug. 1971.
- [Melsa78] J. L. Melsa and D. L. Cohn. *Decision and Estimation Theory*. McGraw-Hill, New York, 1978.
- [Mo98] L. Mo, X. Song, Y. Zhou, Z. Sun, and Y. Bar-Shalom. Unbiased Converted Measurements in Tracking. *IEEE. Trans. Aerospace and Electronic Systems*, AES-34:1023–1027, July 1998.
- [Moghaddamjoo86] A. Moghaddamjoo and R. L. Kirlin. Robust Adaptive Kalman Filtering with Unknown Inputs. In *Proc. 1986 American Control Conf.*, Seattle, WA, June 1986.
- [Moose73] R. L. Moose and P. L. Wang. An Adaptive Estimator with Learning for a Plant Containing Semi-Markov Switching Parameters. *IEEE Trans. Systems, Man and Cybernetics*, SMC-3:277–281, May 1973.
- [Moose75] R. L. Moose. An Adaptive State Estimator Solution to the Maneuvering Target Problem. *IEEE Trans. Automatic Control*, AC-20(3):359–362, June 1975.
- [Moose79] R. L. Moose, H. F. VanLandingham, and D. H. McCabe. Modeling and Estimation of Tracking Maneuvering Targets. *IEEE Trans. Aerospace and Electronic Systems*, AES-15(3):448–456, May 1979.
- [Moose80] R. L. Moose, H. F. VanLandingham, and D. H. McCabe. Application of Adaptive State Estimation Theory. In *Proc. 19th IEEE Conf. Decision and Control*, Albuquerque, NM, Dec. 1980.
- [Mutambbara98] A. G. O. Mutambbara. *Decentralized Estimation and Control for Multisensor Systems*. CRC Press, Boca Raton, LA, 1998.

- [Myers76] K. A. Myers and B. D. Tapley. Adaptive Sequential Estimation with Unknown Noise Statistics. *IEEE Trans. Automatic Control*, AC-21:520–523, Aug. 1976.
- [Nardone81] S. C. Nardone and V. J. Aidala. Observability Criteria for Bearings-Only Target Motion Analysis. *IEEE Trans. Aerospace and Electronic Systems*, AES-17:162–166, March 1981.
- [Navarro77] A. M. Navarro. General Properties of Alpha-Beta and Gamma Tracking Filters, 1977. Report PHL 1977-02, National Defense Research Organization, Physics Lab., The Hague, The Netherlands, Jan. 1977 (US Dept. of Commerce, National Technical Information Services, Report 77-24347).
- [Ng83] L. C. Ng and R. A. LaTourette. Equivalent Bandwidth of a General Class of Polynominal Smoothers. *J. Acoustical Society of America*, 74:814–816, Sept. 1983.
- [O'Halloran88] D. R. O'Halloran and R. K. Baheti. Fast Mapping of a Kalman Filter on WARP, GE-CRD, Schenectady, NY, 1988.
- [Pachter87] M. Pachter. Comments on “Analytical Steady-State Solution for a Continuous Time Kalman Filter”. *IEEE Trans. Aerospace and Electronic Systems*, AES-23:596–597, July 1987.
- [Papoulis84] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, 1984.
- [Parkinson96] B. W. Parkinson and J. J. Spilker (Eds.). *Global Positioning System: Theory and Applications*. AIAA, 1996.
- [Pattipati83] K. R. Pattipati and N. R. Sandell. A Unified View of State Estimation in Switching Environments. In *Proc. 1983 American Control Conf.*, pages 458–465, 1983.
- [Pitman62] G. R. Pitman (Ed.). *Inertial Guidance*. Wiley, New York, 1962.
- [Poor88] H. V. Poor. *An Introduction to Signal Detection and Estimation*. Springer-Verlag, New York, 1988.
- [Press92] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, 1992.
- [Raiffa72] H. Raiffa and R. Schlaifer. *Applied Statistical Decision Theory*. M.I.T. Press, Cambridge, MA, 1972.
- [Rankin94] J. Rankin. GPS and Differential GPS: An Error Model for Sensor Simulation. In *Proceedings of 1994 IEEE Position, Location, and Navigation Symposium*, Las Vegas, NV, April 1994.
- [Rauch65] H. E. Rauch, F. Tung, and C. T. Striebel. Maximum Likelihood Estimation of Linear Dynamic Systems. *AIAA Journal*, 3:1445–1450, Aug. 1965.
- [Rhodes71] I. B. Rhodes. A Tutorial Introduction to Estimation and Filtering. *IEEE Trans. Automatic Control*, AC-16(6):608–704, Dec. 1971.
- [Rogers88] S. R. Rogers. Steady State Performance of Decoupled Kalman Filter. In *Proc. NAECON*, Dayton, OH, May 1988.
- [Sage71] A. Sage and J. Melsa. *Estimation Theory with Applications to Communications and Control*. McGraw-Hill, New York, 1971.
- [Schmidt93] G. C. Schmidt. Designing Nonlinear Filters Based on Daum's Theory. *Journal of Guidance, Control, and Dynamics*, 16(2):371–376, March-April 1993.
- [Schweppe73] F. Schweppe. *Uncertain Dynamic Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1973.
- [Singer70] R. A. Singer. Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets. *IEEE Trans. Aerospace and Electronic Systems*, AES-6:473–483, July 1970.
- [Sittler64] R. W. Sittler. An Optimal Data Association Problem in Surveillance Theory. *IEEE Trans. Mil. Electron.*, MIL-8:125–139, April 1964.
- [Sklansky57] J. Sklansky. Optimizing the Dynamic Parameters of a Track-While-Scan System. *RCA Review*, 18:163–185, June 1957.

- [Song85] T. L. Song and J. L. Speyer. A Stochastic Analysis of a Modified Gain Extended Kalman Filter with Applications to Estimation with Bearings Only Measurements. *IEEE Trans. Automatic Control*, AC-30:940–979, July 1985.
- [Sorenson85] H. W. Sorenson (Ed.). *Kalman Filtering: Theory and Application*. IEEE Press, New York, 1985.
- [Sorenson88] H. W. Sorenson. Recursive Estimation for Nonlinear Dynamic Systems. In J. C. Spall (Ed.), *Bayesian Analysis of Time Series and Dynamic Models*, chapter 6. Marcel Dekker, Inc., New York, 1988.
- [Spingarn72] K. Spingarn and H. L. Weidemann. Linear Regression Filtering and Prediction for Tracking Maneuvering Aircraft Targets. *IEEE Trans. Aerospace and Electronic Systems*, AES-8:800–810, Nov. 1972.
- [Stallard87] D. V. Stallard. An Angle-Only Tracking Filter in Modified Spherical Coordinates. In *Proc. 1987 AIAA Guid., Navig. and Control Conf.*, pages 542–550, 1987.
- [Striebel65] C. Striebel. Sufficient Statistics in the Optimum Control of Stochastic Systems. *J. Math. Anal. and Appl.*, 12:576–592, Dec. 1965.
- [Swerling59] P. Swerling. First Order Error Propagation in a Stagewise Smoothing Procedure for Satellite Observations. *J. Astronautical Sciences*, 12:46–52, Autumn 1959.
- [Sworder89] D. D. Sworder and R. G. Hutchins. Image-Enhanced Tracking. *IEEE Trans. Aerospace and Electronic Systems*, AES-25:701–710, Sept. 1989.
- [Sworder99] D. D. Sworder and J. E. Boyd. *Estimation Problems in Hybrid Systems*. Cambridge University Press, New York, 1999.
- [Tenney77a] R. R. Tenney, T. B. Ballard, and L. E. Miller. A Dual Passive Tracking Filter for Maneuvering Targets. In *Proc. ONR Passive Target Tracking Conf.*, Naval Postgrad. School, Monterey, CA, May 1977.
- [Tenney77b] R. R. Tenney, R. S. Hebbert, and N. R. Sandell. A Tracking Filter for Maneuvering Sources. *IEEE Trans. Automatic Control*, AC-22:246–251, April 1977.
- [Thorp73] J. S. Thorp. Optimal Tracking of Maneuvering Targets. *IEEE Trans. Aerospace and Electronic Systems*, AES-9:512–519, July 1973.
- [Tichavsky98] P. Tichavsky, C. H. Muravchik, and A. Nehorai. Posterior Cramer-Rao Bounds for Discrete-Time Nonlinear Filtering. *IEEE Trans. Signal Processing*, SP-46:1386–1396, May 1998.
- [Tugnait79] J. K. Tugnait and A. H. Haddad. A Detection-Estimation Scheme for State Estimation in Switching Environments. *Automatica*, 15(4):477–481, July 1979.
- [Tugnait82] J. K. Tugnait. Detection and Estimation for Abruptly Changing Systems. *Automatica*, 18(5):607–615, Sept. 1982.
- [Van Trees68] H. L. Van Trees. *Detection, Estimation, and Modulation Theory, Part I*. John Wiley & Sons, New York, 1968.
- [Wang99] H. Wang, T. Kirubarajan, and Y. Bar-Shalom. Precision Large Scale Air Traffic Surveillance Using an IMM Estimator with Assignment. *IEEE Trans. Aerospace and Electronic Systems*, AES-35(1):255–266, Jan. 1999.
- [Weiss80] H. Weiss and J. B. Moore. Improved Extended Kalman Filter Design for Passive Tracking. *IEEE Trans. Automatic Control*, AC-25:807–811, Aug. 1980.
- [Willsky76a] A. S. Willsky. A Survey of Design Methods for Failure Detection in Dynamic Systems. *Automatica*, 12(6):601–611, Nov. 1976.
- [Willsky76b] A. S. Willsky and H. L. Jones. A Generalized Likelihood Ratio Approach to the Detection and Estimation of Jumps in Linear Systems. *IEEE Trans. Automatic Control*, AC-21:108–112, Feb. 1976.
- [Wishner70] R. P. Wishner, R. E. Larson, and M. Athans. Status of Radar Tracking Algorithms. In *Proc. Symp. Nonlinear Estimation*, San Diego, CA, Sept. 1970.

- [Woolfson85] M. S. Woolfson. An Evaluation of Manoeuvre Detector Algorithms. *GEC J. of Research (Chelmsford, England)*, 3(3):181–190, 1985.
- [Yeddanapudi97] M. Yeddanapudi, Y. Bar-Shalom, and K. R. Pattipati. IMM Estimation for Multitarget-Multisensor Air Traffic Surveillance. *IEEE Proceedings*, 85(1):80–94, Jan 1997.
- [Zadeh63] L. A. Zadeh and C. A. Desoer. *Linear System Theory*. McGraw-Hill, New York, 1963.
- [Zhang98] Y. M. Zhang and X. R. Li. Detection and diagnosis of sensor and actuator failures using IMM estimator. *IEEE Trans. Aerospace and Electronic Systems*, AES-34:1293–1313, Oct. 1998.



# Index

- a posteriori, 91  
a priori, 91  
acceleration increment, 274  
acceleration increment over a sampling period, 289  
acceleration, longitudinal and lateral, 418  
accelerometer, 495  
acceptable estimation error, 233  
acceptance interval, 235  
acceptance region, 76  
acceptance regions, 240  
actual MSE, 233  
adaptive bandwidth, 479  
adaptive estimation algorithms, 421  
adjoint, 359  
adjugate, 22  
affine transformation, 362  
air traffic control (ATC), 423, 466  
algebraic matrix Riccati equation, 211, 350  
algebraic Riccati equation, 211  
algorithm comparison, 86  
alpha filter, 296  
alpha filter and suboptimality, 297  
alpha filter, bias and total MS error, 297  
alpha-beta, 278  
alpha-beta filter, 282, 286, 287  
alpha-beta filter design and evaluation, 296  
alpha-beta filter gain, 278  
alpha-beta filter, bias and total MS error, 298  
alpha-beta-gamma, 278  
alpha-beta-gamma filter, 289  
alternate hypothesis, 72  
alternative derivation of the Kalman filter gain, 263  
Altitude estimation from slant range, 118  
amplitude estimation of a sinusoid, 368  
angle-only measurements, 137  
angular rate, 183  
angular rate estimation, 416  
arbitrary gain, 262  
artificial noise, 243  
artificial process noise, 212, 395, 482  
asymptotically efficient, 113  
asymptotically Gaussian, 113  
asymptotically unbiased, 101, 113  
ATC, 423  
augmented state, 481  
augmented state equation, 322  
autocorrelated measurement noise, 327  
autocorrelated noise, 67  
autocorrelated process noise, 320  
autocorrelation, 62  
autocovariance, 62  
autocovariance of the state of a discrete time system, 197  
autocovariance of the state of a system 197  
average, 37  
average covariance associated with the optimal estimate, 378  
average NEES, 234, 238  
average NIS, 236, 238  
average, fading memory, 61  
axioms of probability, 31  
azimuth, 397  
backward recursion, 334  
bandwidth of an alpha filter, 298  
bandwidth of the filter, 207  
bank of filters, 442  
base state, 444, 481  
batch estimator, 130  
Bayes' criterion, 48  
Bayes' decision rule, 48  
Bayes' formula, 48, 91, 114  
Bayes' formula for random variables, 48  
Bayes' postulate, 49  
Bayes' principle, 48  
Bayes' risk, 48  
Bayes' rule, 48  
Bayes' theorem, 48  
Bayesian approach, 91, 114  
bearing, 397  
bearings-only target localization, 176  
best linear state estimator, 207  
best linear unbiased estimator, 126  
bias and total MS error in the alpha filter, 297  
bias and total MS error in the alpha-beta filter, 298  
bias in the measurements, 263  
bias significance, 400  
block processing, 308  
C/A code, 504  
CA model, 272, 277  
calculation of the merging probabilities, 451  
calculation of the mixing probabilities, 455  
carrier phase measurement, 507

- Cartesian basis vector, 383  
 cascaded filtering, 528  
 causal, 68  
 cdf, 33  
 central limit theorem (CLT), 71  
 centralized estimation, 527  
 Chapman-Kolmogorov equation, 87, 374  
 characteristic function, 42  
 Chebyshev inequality, 85  
 chi distribution, 59  
 chi-square distribution with  $n$  degrees of freedom, 58  
 chi-square test, 235  
 Cholesky factor, 211  
 Cholesky factorization, 21, 311  
 closed-loop system matrix, 361  
 CLT, 71  
 colored noise, 67, 320  
 column vectors, 19  
 complementary distribution, 82  
 complementary event, 31  
 complementary filtering, 492  
 completely controllable, 28, 30  
 completely observable, 28, 30  
 completion of the squares, 53  
 composite hypothesis, 75  
 computational requirements, 210  
 condition number, 26, 302  
 conditional covariance matrix, 123, 201  
 conditional expectation, 50  
 conditional mean, 99, 123  
 conditional MSE, 104  
 conditional pdf, 44  
 conditional pdf of the sum of two Gaussian random variables, 88  
 conditional probability, 44  
 conditional variance, 104  
 conditional variance versus unconditional variance, 118  
 confidence level, 74  
 confidence region, 76, 166  
 consistency checking, 261  
 consistency criteria of a filter, 234  
 consistency evaluation, 233  
 consistency in the mean square sense, 108  
 consistency testing, 244, 387  
 consistency tests, 237  
 consistent, 234  
 consistent estimator, 108, 115  
 constant acceleration (CA) model, 272, 277  
 constant acceleration motion, 151  
 Constant altitude trajectory estimation from range-only observations, 175  
 constant angular rate, 183  
 constant jerk motion, 151  
 constant speed, 183  
 constant velocity (CV) model, 270, 274, 276  
 constant velocity motion, 150  
 constant velocity object, 269  
 continuation, track, 153  
 continuous noise level adjustment, 422, 424  
 continuous time state estimate, 346  
 continuous time state estimation filter, 346  
 continuous-discrete filter, 347, 358  
 continuous-time EKF, 416  
 continuous-time innovation, 347  
 continuous-time linear stochastic systems, 180  
 continuous-time linear systems, 27  
 continuous-time nonlinear systems, 404  
 controllability condition, 481  
 controllability matrix, 28, 30  
 convergence in mean square, 108  
 converted measurement, 390  
 convolution, 41  
 coordinate transformation from polar to Cartesian, 415  
 coordinated turn, 183  
 coordinated turn (CT), 467  
 coordinated turn transition matrix, 198  
 correction term, 134  
 correlated noises, 116  
 correlation between current estimate and smoothed value, 339  
 correlation coefficient, 39  
 correlation matrices, 250  
 cost function, 99  
 cost functional, 358  
 cost weighting matrices, 359  
 countable additivity, 31  
 course RMS errors, 243  
 covariance associated with the estimate, 201  
 covariance matching, 233  
 covariance matrices, filter-calculated, 249  
 covariance matrix, 42, 128  
 covariance matrix of the LS estimator, 131  
 covariance of the residual, 133, 134, 173  
 covariance of the state versus covariance of the estimation error, 262  
 covariance of two quadratic forms, 55  
 covariance of two scalar random variables, 39  
 covariance prediction equation, 301  
 covariance propagation equation, 191  
 covariance update, 206  
 covariance update and gain calculation, 314  
 covariance update equation, 302  
 cpmf, 35  
 Cramer-Rao lower bound (CRLB), 109, 164  
 CRLB, 109, 115, 164  
 CRLB as the actual covariance, 169  
 CRLB for a two-model parameter estimation problem, 485  
 cross-correlated noise sequences, 325  
 cross-covariance, 122, 362  
 CT, 467  
 CT model, 467  
 cubic fit, 151  
 cumulative probability distribution function (cdf), 33

- cumulative probability mass function (cpmf), 35  
 cumulative standard Gaussian distribution, 35  
 curse of dimensionality, 380  
 CV model, 270, 274, 276  
 CWNA model, 286  
 data fusion, 524  
 debiasing threshold, 488  
 decision, 1  
 decision-free, 466  
 decorrelation time, 321  
 degree of belief, 48  
 degrees of freedom, 158  
 delta function, 35, 69  
 delta range, 507  
 density, 33  
 derivative of the inverse of a matrix, 367  
 derivative with respect to a matrix, 87  
 design and evaluation, 296  
 design parameter, 267  
 detection probability, 72  
 determinant, 22  
 DFT, 193  
 DGPS, 509  
 diagonal matrix, 21  
 differenced measurement, 327  
 Differential GPS (DGPS), 509  
 diffuse, 49, 117, 137  
 diffuse pdf, 95  
 diffuse prior, 302, 307  
 diffuse uniform prior pdf, 95  
 diffusion process, 180, 410  
 dilution of precision (DOP), 507  
 dimension of a pdf, 86  
 dimensionality, curse of, 380  
 Dirac (impulse) delta function, 35, 65  
 direct discrete time model, 189  
 Dirichlet prior, 49  
 discounting, 396  
 discrete time gain, 187  
 discrete white noise acceleration model, 273  
 discrete white noise acceleration (DWNA) model, 280  
 discrete Wiener process acceleration model, 274  
 discrete-time (difference) matrix Riccati equation, 211  
 discrete-time Fourier transform (DTFT), 193  
 discrete-time linear stochastic systems, 188  
 discrete-time Markov process, 69  
 discrete-time measurement, 345  
 discrete-time measurement equation, 188  
 discrete-time process noise, 187  
 discrete-time stochastic process, 69  
 discrete-time systems, 187  
 discrete-time transfer function, 193  
 discrete-time white noise, 69  
 discrete-time Wiener process, 69  
 discrete-valued parameter, 116  
 discretized continuous-time white noise acceleration (CWNA) model, 286  
 distance estimation in sailing, 419  
 distributed estimation fusion, 528  
 distribution, 33  
 distribution-free probability region, 85  
 divergence, 232  
 divergence of the filter, 385  
 DOP, 507  
 downcrossing threshold, 426  
 drift, 180, 410  
 DTFT, 193  
 dual problems, 360, 361  
 duality, 305  
 DWNA model, 273, 280  
 DWPA model, 274  
 dyad, 26  
 dyad of a state estimate — the expected value, 339  
 DynaEst, 219, 460  
 dynamic equation, 27, 180  
 dynamic estimation problem, 91  
 dynamic MM estimator, 441  
 dynamic programming, 407  
 dynamic range capability, 479  
 effective window length, 425, 432  
 efficiency check, 165  
 efficient, 109  
 efficient estimator, 115  
 eigenvalues, 25  
 eigenvectors, 25  
 EKF, 371, 382  
 EKF prediction equations for a system with unknown parameters, 486  
 EKF simulation for state and parameter estimation, 486  
 ellipsoid,  $g$ -sigma, 166  
 ensemble, 62  
 ensemble average, 62  
 ensemble averages, 64, 237  
 ergodic, 64  
 error state of an inertial navigation system, 497  
 estimability, 166  
 estimate, 90, 122  
 estimate and covariance combination, 457  
 estimate of a parameter, 114  
 estimate of the state, 201  
 estimation, 1  
 estimation criteria that lead to the conditional mean, 117  
 estimation error, 78, 90, 101, 125, 201, 233  
 estimation from indirect observations, 175  
 estimation of the probability of an event, 107  
 estimation with correlated measurements, 119  
 estimation with correlated noises, 172  
 estimation, dynamic, 91  
 estimation, static, 91  
 estimator, 90  
 estimator design parameters, 446  
 estimator of a parameter, 114

- estimator optimality, 199, 233  
 evaluation of estimator optimality, 199, 233  
 evaluation of the CRLB, 113  
 event, 31  
 evidence from the data, 48, 91  
 exhaustive, 45  
 expected value, 37  
 expected value of a quadratic form, 54  
 expected value of a quartic form, 55  
 exponential pdf, 94  
 exponentially autocorrelated acceleration model, 271  
 exponentially autocorrelated noise, 321  
 exponentially decaying autocorrelation, 186  
 exponentially discounted average, 61, 192, 425  
 extended Kalman filter (EKF), 371, 382  
 extrapolation, 152
- factorization, 186, 193  
 factorization of the covariance matrix, 303  
 factorization, Cholesky, 21  
 factorization, LDL $'$ , 21  
 factorization, LDU, 21  
 factorized prediction covariance, 313  
 fading memory, 396  
 fading memory average, 61, 192, 425, 432, 486  
 fading memory average and effective window length, 197  
 fading memory sum, 61, 192  
 false alarm, 72  
 false alarm probability, 73  
 fault detection, 487  
 filter design parameter, 267  
 filter for CWPA model without initial estimates, 318  
 filter for DWNA model without initial estimates, 318  
 filter for DWPA model without initial estimates, 318  
 filter gain, 204, 205, 346  
 filter stability, 212  
 filter tuning, 243  
 filter-calculated covariance, 233  
 filter-calculated covariance matrices, 249  
 filter-calculated covariances, 207  
 filtering, 2  
 FIM, 110, 115, 164  
 finite impulse response (FIR), 194  
 finite-sample consistency, 234  
 FIR, 194  
 first moment, 37  
 first-order polynomial fitting, 247  
 Fisher approach, 91  
 Fisher information, 109  
 Fisher information matrix, 115  
 Fisher information matrix (FIM), 110, 164  
 Fisher information matrix in bearings-only target localization, 173  
 Fisher information matrix in the LS problem, 173  
 fitting error, 155–157, 161, 167  
 fixed interval prediction, 330, 356  
 fixed lead prediction, 330
- fixed lead prediction evolution in continuous time, 368  
 fixed point prediction, 330  
 fixed point prediction — the mean and covariance, 338  
 fixed point prediction evolution in continuous time, 367  
 fixed point prediction prior moments, 368  
 fixed-interval prediction, 332  
 fixed-interval smoothing, 333, 334  
 fixed-lag smoothing, 333  
 fixed-lead prediction, 332, 356  
 fixed-point prediction, 356  
 fixed-point smoothing, 333  
 Fokker-Planck (F-P) equation, 412  
 Fokker-Planck equation, 410  
 forward diffusion equation, 410  
 fourth joint moment of Gaussian random variables, 87  
 frequency estimation of a sinusoid, 368  
 fudge factor, 396, 424  
 full rank, 25, 28  
 function in the interval, 181  
 fundamental equations of linear estimation, 54, 128  
 fusion equations, 529  
 fusion of estimates, 264
- gain, 134  
 gain coefficients, 289  
 gate, 153, 154  
 gate probability, 166  
 Gating, 263  
 Gauss-Markov sequence, 69  
 Gaussian, 51  
 Gaussian assumption as the worst case, 128  
 Gaussian mixture, 55, 445  
 Gaussian random variable, 34  
 GDOP, 145, 508, 531  
 generalized pseudo-Bayesian (GPB), 423, 446  
 generalized pseudo-Bayesian approach of first order (GPB1), 447  
 generalized pseudo-Bayesian estimator of second order (GPB2), 449  
 generation of a random sequence with exponential autocorrelation, 317  
 generation of correlated random variables, 317  
 Geometric Dilution of Precision, 145  
 Geometric Dilution of Precision (GDOP) in Global Positioning System (GPS), 417  
 Geometric Dilution of Precision in GPS, 531  
 Geometric DOP (GDOP) matrix, 508  
 gimbaled system, 495  
 Global Positioning System, 176, 417  
 good observability, 167  
 goodness-of-fit, 155, 156  
 GPB, 423, 446  
 GPB1, 447  
 GPB2, 449

- GPS, 176, 417, 501  
 GPS-aided INS, 524  
 gradient, 24  
 Gram-Schmidt orthogonalization, 313, 317  
 guidance, 491  
 gyroscope, 495
- Hessian, 24  
 hidden Markov model, 445  
 higher-order polynomial fitting, 247  
 higher-order terms, 383  
 homogeneous Markov chain, 445  
 hybrid, 441  
 hybrid random variable, 36  
 hybrid system with i.i.d. modal state sequence, 487  
 hybrid systems, 48, 57, 423  
 hypothesis testing, 72  
 hypothesis testing with correlated noise, 87
- i.i.d., 41, 71  
 i.i.d. modal state sequence in a hybrid system, 487  
 idempotent, 21, 25  
 identity matrix, 22  
 IE, 436  
 iff, 21  
 ILS estimator, 138  
 IMM, 423, 446  
 IMM estimator, 453  
 implementation of the linear estimation techniques, 301  
 imported measurements, 223  
 impossible event, 31  
 improper densities, 34  
 improper pdf, 95  
 incompletely observed Markov process, 376  
 incorrect gain, 206  
 incorrect prior variance, 173  
 independence of measurements, 198  
 independence versus conditional independence, 85  
 independent, 41  
 independent increment process, 66  
 independent runs, 237  
 independent, identically distributed (i.i.d.), 41, 71  
 inequality of two matrices, 26  
 inertial navigation system (INS), 495  
 inertial navigation system, gimbaled, 495  
 inertial navigation system, strapdown, 495  
 information, 94, 132  
 information extraction and enhancement, 2  
 information filter, 248, 302, 304  
 information filter state, 307  
 information in the data, insufficient, 165  
 information limits, 2  
 information matrix, 304, 349  
 information matrix filter, 206  
 information set, 373  
 information state, 181, 373, 375  
 information, existing, 110
- information, extracted, 110  
 information, new, 146  
 information, old, 145  
 initial covariance, 202  
 initial estimate, 202  
 initial information, 202  
 initialization, 134, 147, 245, 262  
 inner product, 20, 124  
 innovation, 205  
 innovation covariance, 205, 384  
 innovation sequence, 213  
 innovations representation, 214  
 input estimate, 430  
 input estimation, 422  
 input estimation (IE), 436  
 input gain, 27, 180  
 input vector, 180  
 INS-aided GPS, 524  
 INS/GPS integration by distributed estimation fusion, 528  
 insignificant, 77  
 instantaneous variance, 65, 321  
 insufficient information in the data, 165  
 integrated navigation, 524  
 integration of INS and GPS, 524  
 intensity of the white noise, 342  
 interacting multiple model (IMM), 423, 446  
 interacting multiple model (IMM) estimator, 453  
 interaction, 454  
 interpolating polynomial, 182  
 interval estimate, 77  
 inverse, 21  
 inversion lemma, 23  
 inversion, partitioned matrix, 22  
 iterated extended Kalman filter, 406  
 Iterated Least Squares estimator, 138  
 Ito differential equation, 343, 347
- Jacobian matrix, 24  
 jerk, 151, 271  
 joint cdf, 39  
 joint density, 41  
 joint pdf, 38  
 joint probability-pdf, 36  
 jointly Gaussian, 51, 52  
 Joseph form covariance update, 206, 302, 310  
 jump probabilities, 70  
 jump process, mode, 444  
 jump-linear systems, 444
- Kalman filter, 199  
 Kalman filter (KF), 202  
 Kalman filter gain, 263  
 Kalman filter with different measurements, 265  
 Kalman filter with different sampling intervals, 265  
 Kalman filter with nonzero noise means, 263  
 Kalman-Bucy filter, 341, 346  
 Kalman-Bucy filter with non-zero-mean noises, 369

- KF, 202  
 KF as a lowpass filter, 369  
 kinematic state models, 268  
 known acceleration objects, 299  
 known input, 200  
 Kolmogorov forward equation, 410, 412  
 Kronecker delta function, 69
- Laplacian pdf, 88, 117  
 large-sample properties of the ML estimator, 113  
 law of iterated expectations, 50  
 law of large numbers (LLN), 70  
 LDL' factorization, 21, 311, 312  
 LDU factorization, 21  
 least mean square (LMS), 115, 129  
 least squares (LS), 129  
 least squares (LS) method, 98  
 least squares estimate, 76  
 least squares estimator (LSE), 98  
 left half-plane (LHP), 364  
 left-continuous, 444  
 length of a vector, 23  
 level of significance, 74  
 LF, 91  
 LG, 201  
 likelihood equation, 92, 98  
 likelihood equations, 102  
 likelihood function, 48, 73, 97, 114, 374  
 likelihood function (LF), 91  
 likelihood function of mode, 442  
 likelihood function of the system model, 214  
 likelihood ratio, 73  
 limiting process of an autocorrelated noise to white noise, 338  
 linear dependence, 124  
 linear Gaussian (LG) assumption, 201  
 linear least squares (LS), 129  
 linear LS problem, 98  
 linear MMSE, 124  
 linear MMSE estimator, 125, 129  
 linear MMSE estimator for the multidimensional case, 127  
 linear space, 124  
 linear time-invariant system, 27, 30  
 linear transformation of random variables, 87  
 linear-quadratic control problem (LQ), 358  
 linearly dependent, 40  
 linearly independent, 22  
 lines of sight (LOS), 507  
 LLN, 70  
 LMMSE estimation for vectors, 174  
 LMMSE estimator, 176  
 LMMSE state estimator, 207  
 LMS, 115, 129  
 log-likelihood function, 98, 164  
 log-normal random variable, 86  
 LOS, 507  
 loss of positive definiteness, 301  
 loss of symmetry, 301  
 low probability events, 86  
 lower triangular matrix, 21, 311  
 lowpass filter, 369  
 LQ, 358  
 LS, 98, 114, 129  
 LS estimator, 130  
 LSE, 98  
 lumped estimate, 447  
 Lyapunov differential equation, 347  
 Lyapunov equation, 185
- MA, 194  
 magnitude of a vector, 23  
 Mahalanobis distance, 166  
 maneuver, 466  
 maneuver detection, 430  
 maneuver onset, 433  
 maneuver onset time, 428  
 maneuver termination, 433  
 maneuvering index, 267, 281, 287  
 maneuvering index threshold, 479, 488  
 maneuvering model, 432  
 MAP, 92  
 MAP estimate, 114  
 MAP estimate with two-sided exponential (Laplacian) prior, 117  
 MAP smoothing technique, 409  
 MAP with Gaussian prior — vector version, 118  
 margin of error, 108  
 marginal density, 39  
 marginal observability, 167  
 marginal pdf, 39  
 Markov chain, 70  
 Markov chain Monte Carlo methods, 414  
 Markov inequality, 85  
 Markov process, 184, 190  
 Markov processes, 66  
 Markov property, 66  
 Markov sequence, 69, 190  
 matched, 237  
 matrix, 19  
 matrix inversion lemma, 23  
 matrix MSE, 128  
 matrix Riccati differential equation, 346, 349  
 maximization, multistage, 407  
 maximum a posteriori (MAP), 92  
 maximum likelihood estimate, 76  
 maximum likelihood estimator (MLE), 92  
 maximum likelihood method, 92  
 Maxwell distribution, 59  
 MCMC methods, 414  
 mean, 37, 62  
 mean and covariance of the state of a linear system, 196  
 mean sojourn time, 460  
 mean sojourn time, 471  
 mean square (MS), 37

- mean square error (MSE), 104  
 mean square error (MSE) matrix, 140  
 mean-square value matrix, 250  
 measure, 31  
 measure of belief, 32  
 measurement, 122  
 measurement equation, 27, 180  
 Measurement error variance, 119  
 measurement matrix, 27, 180, 188  
 measurement noise, 180, 188, 200  
 measurement prediction covariance, 203, 205, 384  
 measurement prediction error, 205  
 measurement residual, 205  
 measurement update, 208  
 measurement-dependent accuracy, 377  
 measurements, 3  
 measurements, imported, 223  
 measurements, simulated, 223  
 median, 117  
 merging, 452  
 merging probabilities, 451  
 minimum magnitude error estimate, 117  
 minimum mean square error (MMSE), 123  
 minimum mean square error (MMSE) estimator, 99  
 minimum variance (MV), 115, 129  
 mismatched, 237  
 mismatched filter, 240  
 miss, 72  
 miss probability, 74  
 mixed coordinate filter, 390  
 mixed pmf/pdf model — the variance, 338  
 mixed pmf/pdf of the acceleration model, 339  
 mixed random variable, 36  
 mixing, 454, 455  
 mixing probabilities, 454, 455  
 mixture, 55, 86  
 mixture equations, 56  
 ML estimate, 114  
 MLE, 92  
 MLE from correlated measurements, 118  
 MM, 423, 441  
 MM estimator, dynamic, 441  
 MM estimator, static, 441  
 MMSE, 114, 123  
 MMSE estimation with correlated noises, 172  
 MMSE estimator, 99, 123  
 MMSE with exponential prior, 118  
 Mobile sensor platform location error, 531  
 Mobile sensor platform location error reduction, 532  
 modal state, 444  
 modal trajectory, 371, 407  
 mode, 93, 100, 407, 441  
 mode (model) switching, 444  
 mode conditioned covariances, 442  
 mode conditioned estimate, 449  
 mode conditioned state estimate, 442  
 mode conditioned state estimates, 442  
 mode estimates, 442  
 mode history, 444  
 mode jump, 443  
 mode jump process, 444  
 mode jumping, 441  
 mode likelihood function, 442, 447  
 mode matched filter, 442  
 mode observation in a hybrid system, 487  
 mode probabilities, 452  
 mode probability update, 448, 456  
 mode probability updating, 452  
 mode transition probabilities, 445, 446, 452  
 mode-conditioned state estimate, 447  
 mode-matched filtering, 448, 451, 456  
 model, 441  
 modes, 423  
 modified process noise covariance, 396  
 moment, 37  
 moment matching, 56, 60  
 moment-generating function, 43  
 moments of the white noise response of a scalar linear system, 196  
 Monte Carlo runs, 168, 439  
 Monte Carlo runs for algorithm comparison, 86  
 Monte Carlo runs for low probability events, 86  
 Monte Carlo simulations, 168  
 Monte Carlo simulations (runs), 79, 234  
 motion uncertainty, 281  
 moving average (MA), 194, 425, 433  
 moving sum, 425  
 MS, 37  
 MSE, 104, 233  
 MSE matrices of predicted and updated state, 250  
 MSE matrix, 140, 384  
 MSE, unconditional, 104  
 MSV matrix, 250  
 multiframe signal processing, 409  
 multilateration, 137  
 multiple model (MM), 423  
 multiple model (MM) approach, 441  
 multiscan signal processing, 409  
 multisensor, 309  
 multistage maximization, 407  
 mutually exclusive, 31, 45  
 MV, 115, 129  
 navigation, 2, 491  
 Navigation Satellite Timing and Ranging/Global Positioning System --- Navstar/GPS, 501  
 NCA model, 272  
 NCV model, 270, 274  
 nearly constant acceleration (NCA) model, 272  
 nearly constant velocity model, 270, 274  
 nearly coordinated turn model, 467  
 NEES, 165, 234, 238  
 Neyman-Pearson Lemma, 72  
 NIS, 236  
 NLS, 137, 164  
 NMEE, 235

- noise, 179  
 Noise correlated to prior error, 173  
 noise gain, 180, 189, 272  
 noise level switching, 422, 424  
 noise only measurement, 174  
 noise reduction, 438  
 noise reduction factor, 282, 458  
 noiseless kinematic model, 276  
 nominal trajectory, 385  
 non-Bayesian, 91  
 non-Bayesian approach, 91, 114  
 noncausal, 68, 364  
 noninformative, 49, 137  
 noninformative pdf, 95  
 noninformative prior, 302, 307  
 nonlinear, 164  
 nonlinear least squares, 161, 164, 176  
 Nonlinear Least Squares estimator, 137  
 nonlinear LS, 130  
 nonlinear LS problem, 98  
 nonmaneuvering filter, 427  
 nonsingular, 22, 25  
 nonstationary, 191  
 nonstationary white process noise, 184  
 nonstationary zero-mean white, 65  
 norm, 23, 124  
 norm of a matrix, 27  
 norm of the residual, 156  
 normal (Gaussian), 83  
 normal pdf (density), 51  
 normal random variable, 34, 51  
 normalization constant, 91  
 normalization property, 33, 35  
 normalized (state) estimation error squared (NEES), 234  
 normalized error squared (NES), 401  
 normalized estimation error squared (NEES), 165  
 normalized innovation squared, 424  
 normalized innovation squared (NIS), 236  
 normalized mean estimation error (NMEE), 235  
 normalized sum of the squares of the residuals, 167  
 normalizing constant, 48  
 normally (Gaussian) distributed, 122  
 nowhere differentiable, 66  
 null hypothesis, 72  
 numerical integration, 404  
 observability Gramian, 349  
 observability matrix, 28, 30, 349  
 observability of a linear dynamic system, 266  
 observability, marginal (poor), 132  
 observability, parameter, 165  
 observable, 132  
 observation, 122  
 observation uncertainty, 281  
 one sigma region, 153  
 one-sided alternative hypothesis, 75  
 one-sided exponential pdf, 94  
 one-sided probability region, 82, 83  
 one-step prediction, 152  
 operating regime, 441  
 operator, 375  
 optimal cost, 359  
 optimal cost matrix, 359  
 optimal estimator, 2  
 optimal gain, 207  
 optimal MMSE state estimator, 207  
 optimal nonlinear continuous-discrete filter, 410  
 optimal nonlinear state estimator, 372  
 optimistic, 245  
 Ornstein-Uhlenbeck process, 186, 321  
 orthogonal, 23, 124  
 orthogonal projection, 23, 125  
 orthogonality, 363  
 orthogonality of estimation error to previous estimates, 263  
 orthogonality of the innovations to the state prediction, 263  
 orthogonality properties, 215  
 orthonormal vectors, 158  
 outer product, 20  
 output equation, 27, 180  
 overfitting, 154  
 parabolic fit, 151  
 parameter, 90  
 parameter estimate significance test, 155  
 parameter observability, 165  
 partitioned covariance matrix, 53  
 partitioned matrix inversion, 22  
 passive localization, 161  
 passive localization with direction cosine measurements, 173  
 passive ranging, 161  
 pdf, 33  
 percentile point, 82  
 perversity of inanimate objects, 88, 96  
 pessimistic, 245  
 phase estimation of a sinusoid, 368  
 physical dimension, 344  
 piecewise constant acceleration, 273  
 piecewise constant white acceleration model, 273  
 piecewise constant Wiener process acceleration model, 274, 289  
 plant equation, 27, 180  
 plant noise, 180  
 pmf, 35  
 point estimate, 77  
 point masses, 35  
 point prediction, 374  
 Poisson pmf, 36  
 polar coordinates, 397  
 polar-to-Cartesian conversion bias, 416  
 polynomial filters, 268  
 polynomial fitting, 146, 149  
 polynomial model, 269

- polynomial models, 268  
 position and velocity estimation for known acceleration objects, 299  
 position confidence region, 166  
 position estimate for an arbitrary time, 166  
 position estimation, 175  
 position estimation improvement, 282  
 position gain, 281  
 position RMS errors, 243  
 position/velocity measurement, 297  
 positive (semi)definite matrix, 25  
 positive (semi)definite quadratic form, 25  
 posterior, 91  
 posterior pdf, 48  
 posterior probability, 48  
 power of the test, 72  
 power spectral density, 62  
 power spectral density (spectrum), 193  
 power spectral density matrix of the output, 186  
 power spectrum, 62, 186  
 Precision Positioning Service (PPS), 504  
 predicted measurement, 203, 205, 384  
 predicted state, 202, 204, 355, 383  
 predicted state covariance, 203  
 predicted value, 134  
 predicted value of the state, 201  
 prediction, 152, 330  
 prediction covariance, 356  
 prediction with unequal sampling times and variances, 176  
 predictor-corrector algorithm, 12  
 prewhitening, 67  
 prewhitening of a colored noise, 339  
 prewhitening of a second-order process, 339  
 prewhitening system, 67, 186, 193, 320, 321  
 principle of orthogonality, 124, 125  
 prior, 91  
 prior information, 136  
 prior pdf, 48  
 prior probability, 48  
 probability, 31  
 probability concentration ellipsoid, 166  
 probability concentration region, 235  
 probability density function (pdf), 33  
 probability estimation for an event, 107  
 probability mass, 33  
 probability mass function (pmf), 35  
 probability matrix, 88  
 probability of detection, 74  
 probability region, 76  
 probability region, distribution-free, 85  
 probability region, Gaussian, 83  
 probability region, two-sided, 156  
 process noise, 180, 200, 212  
 process noise and initialization, 263  
 process noise intensity choice, 270, 272  
 process noise rescaling, 295  
 propagation equation of the covariance, 185  
 propagation equation of the mean, 184, 191  
 proper density, 33  
 proper pdf, 95  
 proper pmf, 35  
 pseudo-noise, 212, 243, 395, 482  
 pseudorange equation, 506  
 public opinion polls, 119  
 quadratic form, 25  
 quadratic form with non-zero-mean random variables, 88  
 quadratic observations, 175  
 quartic form, 54  
 quiescent model, 432  
 radar cross-section estimation, 418  
 random number generator testing, 262  
 random process, 61  
 random sequence, 69  
 random variable, 33  
 random walk, 65  
 range, 397  
 range: Horizontal and Slant, 417  
 rank, 25  
 rational function, 186, 193  
 rational spectrum, 67, 186  
 Rayleigh distribution, 59  
 real-time consistency tests, 237, 240  
 realization, 33, 91  
 reciprocal basis, 26  
 recursive LS estimator, 134  
 recursive parameter estimate updating equation, 134  
 reference time, 146, 149, 150  
 regime, operating, 441  
 region, one sigma, 153  
 regression, 146  
 relative frequency, 32  
 relinearization of the measurement equation, 405  
 rescaling of the variance of the process noise, 275  
 residual, 134  
 residual covariance, 384  
 retrodicted state, 201  
 retrodiction, 333  
 revisit rate, 267, 292  
 revisits, 225  
 Riccati differential equation, 346  
 Riccati differential equation solution for the inverse covariance, 367  
 Riccati equation, 211  
 right half-plane (RHP), 364  
 RMS error from Monte Carlo runs, 243  
 RMS speed and velocity errors in Monte Carlo simulations, 417  
 RMS value of a random variable, 37  
 root mean square value of a random variable, 37  
 round-off errors, 301  
 row vector, 19  
 SA, 506

- sample autocorrelation, 236, 238  
 sample average, 71, 79, 100  
 sample correlation, 262  
 sample mean, 79, 100, 135, 439  
 sample performance differences, 81, 439  
 sample variance, 103  
 sampling frequency, 267  
 sampling period change, 295  
 sampling rate, 292  
 scalar updates, 308  
 scaling factor, 424  
 scenario, 246  
 Schwarz inequality, 23, 111  
 second central moment, 37  
 second-order kinematic model, 269  
 Selective Availability, 506  
 sensitivity of the filter to an incorrect gain, 206  
 sensitivity to incorrect prior variance, 173  
 sensitivity to process noise variance, 264  
 sensor bias and target location estimation, 264  
 sensor platform location error, 531  
 sensor platform location error reduction, 532  
 sequence of models, 444  
 sequential maximization, 407  
 sequential Monte Carlo methods, 414  
 sequential processing, 308  
 sequential updating, 303  
 shaping filter, 67, 186, 193, 320, 339  
 short-term average, 345  
 sigma,  $g$ -ellipsoid, 166  
 signal interpolation for measurement generation, 414  
 significance test, parameter estimate, 155  
 significant estimate, 77  
 simulated kinematic trajectory behavior, 295  
 simulated kinematic trajectory variability, 295  
 simulated measurements, 223  
 single run, 237  
 sliding window, 425  
 slowly varying parameters, 482  
 SMC methods, 414  
 smoothed covariance, 336  
 smoothed state, 336  
 smoothed state covariance, 337  
 smoothed value of the state, 201  
 smoother gain, 336  
 smoothing, 333  
 smoothing compared with estimation, 339  
 smoothing property of the expectations, 50  
 smoothing residual, 337  
 soft switching, 458, 466  
 sojourn time, 460, 471  
 sojourn time in a state for a Markov chain, 487  
 spatial window, 154  
 spectral factorization, 67  
 spectral factorization for a discrete-time system, 198  
 spectral factorization for prewhitening (shaping), 197  
 spectral representation, 26  
 speed, 183  
 speed and course estimates, 416  
 speed and velocity errors in Monte Carlo simulations, 417  
 speed RMS errors, 243  
 spread of the means, 443  
 spread of the means term, 56  
 square root, 21  
 square root filtering, 303, 311  
 square root of a matrix, 311  
 squared norm of the estimation error, 233  
 squared norm with respect to a matrix, 26  
 stable, 193, 351  
 stacked measurement matrix, 130  
 stacked vector, 52, 122, 130, 481  
 standard, 83  
 standard deviation, 37  
 standard deviation associated with the estimator, 105  
 standard deviation improvement versus variance improvement, 415  
 standard deviation of the estimate, 78  
 standard deviation of the estimation error, 78, 105  
 standard error, 78, 79, 105, 106  
 standard error of the estimate, 106  
 standard error of the sample variance, 106  
 standard Gaussian distribution, 35  
 standard Gaussian random variables, 58  
 Standard Positioning Service (SPS), 504  
 state, 27, 30, 181  
 state augmentation, 320  
 state equation for fading memory average, 198  
 state estimate and covariance combination, 449, 452  
 state estimation error, 347  
 state estimation errors' autocorrelation, 263  
 state MSE matrix for continuous-time filter with arbitrary gain, 368  
 state prediction covariance, 203, 204, 384  
 state prediction error, 204  
 state prediction in a linear time-invariant discrete time system, 198  
 state prediction pdf, 374  
 state space representation, 187  
 state transition matrix, 181  
 state transition pdf, 375  
 state update, 206  
 state vector, 180  
 state-space representation, 180  
 static estimation problem, 91  
 static MM estimator, 441  
 stationary, 62  
 stationary Markov process, 67  
 stationary white sequence, 69  
 stationary zero-mean white, 65  
 statistical characterization of the disturbances, 233  
 statistically insignificant, 155, 161  
 statistics, 51  
 steady-state covariance, 211, 350  
 steady-state filter, 350, 352

- steady-state filter existence for a kinematic model, 296  
 steady-state filters for noisy kinematic models, 278  
 steady-state gain, 211  
 stochastic observability, 166  
 stochastic process, 61  
 straight line fit, 150  
 strapdown system, 495  
 strict stationarity, 62  
 strict-sense whiteness, 65  
 subspace, 124  
 sufficient statistic, 96  
 superefficiency, 118  
 sure event, 31  
 switching models, 441  
 switching, mode, 444  
 symmetric, 19  
 system design, 267, 292  
 system identification, 421  
 system matrix, 27, 180
- tail mass, 73  
 tail probability, 73  
 target localization, bearings-only, 176  
 target maneuvering index, 267, 281  
 target motion analysis, 161  
 target tracking, 491  
 target tracking index, 281  
 test statistic, 75  
 third-order model, 271  
 three-point initialization, 264  
 tightly coupled, 527  
 time average autocorrelation, 237  
 time average normalized innovation squared, 237, 240  
 time averages, 64, 237  
 time update, 208  
 time-average autocorrelation, 240  
 time-invariant, 190  
 time-invariant system, 27, 30  
 time-varying discrete-time system, 188  
 total probability theorem, 45  
 trace, 20  
 track before detect, 409  
 track continuation, 153  
 tracking, 2  
 tracking index, 281  
 trade-off, 292  
 transfer function matrix, 67  
 transition matrix, 187  
 transition matrix of the Markov chain, 70  
 transition probabilities, 70  
 transpose of a product, 20  
 transposition, 19  
 triangular matrix, 311  
 triangulation, 137  
 truncated Gaussian, 45  
 truncated pdf, 44  
 tuning of the filter, 482
- turn, 183  
 two-model parameter estimation problem with a prior, 486  
 two-point differencing, 247  
 two-sided alternate hypothesis, 76  
 two-sided probability region, 83, 156, 168  
 type I error, 72  
 type II error, 72
- UM, 466  
 unbiased, 101  
 unbiased estimator, 115  
 unbiasedness, 233, 363  
 unbiasedness property, 126  
 unconditional MSE, 104  
 uncorrelated, 40  
 uncorrelatedness, 69  
 underfitting, 154  
 uniform motion (UM), 466  
 uniform prior pdf, 49  
 uniformly distributed random variable, 34  
 unimodal, 101  
 unit lower triangular matrix, 21, 311  
 unit step function, 35  
 unknown input, 427  
 unknown input as a random process, 422  
 unnormalized errors, 243  
 unobservable, 162, 167  
 upcrossing threshold, 426  
 update gain, 133  
 update with an arbitrary gain, 262  
 updated covariance, 206  
 updated state, 203  
 updated state covariance, 203  
 updated state covariance conditioned on prior data, 338  
 updated state estimate, 203, 205  
 upper tail, 82  
 upper tail probability, 82  
 upper triangular matrix, 311
- validity limit of the standard conversion, 400  
 variability reduction, 235  
 variable state dimension, 422  
 variable state dimension (VSD), 431, 436  
 variance, 37  
 variance estimate with the smallest MSE, 117  
 variance of the estimator, 104  
 variance of the sample mean, 106  
 variance of the sample variance, 106  
 variance, conditional, 104  
 variance/MSE of an estimator, 115  
 vector, 19  
 vector Taylor series expansion, 383  
 vector-valued Gaussian, 51  
 vector-valued random process, 62  
 vector-valued random variable, 41  
 vectors in an abstract vector space, 124

velocity estimation, 174  
velocity estimation improvement, 282  
velocity gain coefficient, 281  
velocity increment over a sampling period, 436  
velocity RMS errors, 243  
Viterbi algorithm, 409  
VSD, 431, 436  
  
weak independence, 69  
weighted sum of chi-square variables, 60  
weighted sum of pdfs, 55  
weighting matrix, 127  
WGS-84, 515  
white noise, 64  
white noise acceleration, 268  
white noise acceleration model, 269  
white noise jerk (WNJ) model, 271  
white sequence, 69  
whiteness test, 236  
Wide Area Augmentation System (WAAS), 509  
wide sense stationary, 62  
wide-sense whiteness, 65  
Wiener filter, 364, 365  
Wiener filter as the steady-state Kalman-Bucy filter,  
    369  
Wiener process acceleration, 268  
Wiener process acceleration (WPA) model, 271  
Wiener process increments, 88  
Wiener random process, 65  
Wiener-Hopf equation, 363  
Wiener-Hopf problem, 362  
window length for fading memory average, 197  
window, spatial, 154  
WNA model, 269  
World Geodetic System of 1984 (WGS-84), 515  
WPA model, 271  
  
zero-mean random variables, 124